AN AUTOMATED FRAMEWORK TO SUPPORT AGILE ADOPTION COACHING

by

EMIL JOHANN SCHNABEL

submitted in accordance with the requirements for

the degree of

MASTER OF SCIENCE

in the subject

COMPUTING

at the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR**:** Prof Ernest Mnkandla

December 2018

ABSTRACT

This dissertation aims to assist Agile champions in their organisations with their Agile adoption journey. Such a journey typically starts with identifying the Agile challenges and then determining how to address these challenges in their unique context (Hajjdiab & Taleb, 2011a:9).

This study provides a manual together with an automated framework to showcase a list of prioritised Agile challenges and their corresponding possible solutions.

Before reaching this automated framework, the research presented a comprehensive list of Agile adoption challenges, possible solutions and a mapping between them.

The automated framework, also named Agile Adoption Coach (AAC), is in the form of an online tool: www.agilesense.coza/aac.

The critical quantitative contribution to this study was the evaluation of the AAC by Scrum Masters and Agile coaches to determine the acceptability of the tool. The research has shown that there is, on average, an 86% acceptability rating for this tool in South Africa.

DECLARATION

Name: Emil Johann Schnabel

Student number: 32192770

Degree: MSc in Computing

Exact wording of the title of the dissertation as appearing on the electronic copy submitted for examination:

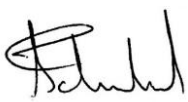
An Automated Framework to Support Agile Adoption Coaching

I declare that the above dissertation is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

I further declare that I submitted the dissertation to originality checking software and that it falls within the accepted requirements for originality.

I further declare that I have not previously submitted this work, or part of it, for examination at Unisa for another qualification or at any other higher education institution.

(The dissertation will not be examined unless this statement has been submitted.)



____ _____ ____31/12/2018__

SIGNATURE DATE

KEYWORDS

Agile, Agile coaching, Agile Challenges, Agile Assessment, Agile Tool, Agile Solutions, Agile Framework, Agile Issues, Agile Adoption, Agile coach, Agile Adoption Coaching

CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

TABLE OF ACRONYMS

4-stage process	Discontinuing Factors, Project-Level Assessment, Organisational Readiness Assessment, Reconciliation
AAC	Agile Adoption Coach
AAIM	Agile Adoption and Improvement Model
CMMI	Capability Maturity Model Index
DSDM	Dynamic Systems Development Method
FDD	Feature-Driven Development
IEEE	Institute of Electrical and Electronic Engineers
IQR	Interquartile Range
ISO	International Standards Organisation
OPP	Objectives, principles, and practices
PMO	Project Management Office
R&D	Research and Development
SAFe	Scaled Agile Framework
SAMI	Sidky Agile Measurement Index
SDM	Software development methodology
TDD	Test Driven Development
UX	User Experience
XP	Extreme Programming
μ	Mean
σ	Standard deviation

STUDY ARTEFACT

Link to Agile Adoption Coach (AAC): www.agilesense.co.za/aac

CITATION MANAGEMENT AND REFERENCE METHOD

References were managed electronically with the Mendeley citation manager. Mendeley offers approximately 7000 citation styles. For consistency, the University of Cape Town Harvard method (UCT Harvard) of referencing was used throughout this dissertation (De Jager & Steele, 2015).

Chapter 1: **INTRODUCTION**

1.1 Introduction and Background

This study did not aim to provide an automated framework in such a way to purport to replace the work of the Agile coach, but rather to augment the work of the Agile coach and act as a tool to fast-track the resulting solutions to challenges found at a company. Many prior studies (Adkins, 2010; Srinivasan & Lundqvist, 2010:130; Kairi, 2011; O'Connor & Duchonova, 2014) found that skilled Agile coaches are essential for any proper Agile adoption.

In the VersionOne's "2015 State of Agile" report, it reported while 33% of respondents said they had five years plus of Agile experience, the organisations within which they worked had less experience (VersionOne, 2015:6). Only 25% of organisations had been using Agile for five or more years (ibid.). The lack of experience indicates that while individuals' Agile adoption is on the up-curve, organisations are still struggling to adopt Agile adequately.

There are two scenarios that can explain the results of the VersioneOne report on individual versus organizational adoption. In scenario one certain pockets in an organization support Agile. Individuals in these pockets are allowed to live out their Agile believes, but the larger organization may still struggle to adopt the values and principles. The danger in these organizations is that the Agile pockets may not last long as there are not sufficient leadership buy-in and support. In scenario two specific individuals in an organization can adopt the Agile philosophy in principle; if the organization do not adopt the values and principles, the individual's adoption cannot be physically realized. For example, a program manager in an insurance firm believes that customer collaboration is vital, but he is not allowed to engage with their brokers regularly in order to validate user experience and requirements for his program. In this case, customer collaboration is a critical Agile value that the organization has not adopted yet.

What does the term "Agile adoption" mean? **Agile** is a mindset and a philosophy (Levy, 2009). The Agile manifesto (Values) and its 12 principles (Beck et al., 2001) describe the essence of Agile. There are also several practices that relate to this (Agile Alliance, 2001); however, the values and principles are the most important. One does not, therefore, need

to follow all practices to be Agile. It is important to note that Agile is not a methodology. There are methodologies like Scrum, Extreme Programming (XP), Kanban and various others that underscore this philosophy.

The word **adoption** in the Agile context refers to organisations or departments adopting this mindset. It is becoming more popular for companies, in general, to adopt Agile and implement the related methodologies (Rodríguez et al., 2012:20; Dingsøyr & Moe, 2013:38). This adoption could, in effect, impact the whole organisation, not only the IT-department, as it speaks about the way one approaches software and all the stakeholders surrounding the software. Qumer and Henderson-Sellers (2008:4) repeatedly refer to the business value Agile can bring to the organisation, and that the impact of Agile adoption is not only on IT. The adoption of Agile then speaks to a change in the philosophy of the organisation.

It has been illustrated that the proper implementation of the Agile philosophy, together with an appropriate framework or methodology to operate in, can increase productivity, team motivation, quality of outcome and generally lead to more successful projects (Chan & Thong, 2009:803; Laanti, Salo & Abrahamsson, 2011:277). It is therefore clear that if more companies were to adopt Agile, it would have a direct positive impact not only on the company but a country's economy as a whole (Conboy, 2009:338-341).

Agile adoption success, however, generally differs tremendously depending on culture, company size, industry and the manner in which Agile is introduced. In the South African context, Agile adoption has been a struggle. According to Weyer (2016:para.1) and Gonçalves (2016), the following factors prevent South African companies from adopting Agile readily, even though the advantages of becoming Agile, are well established:

- Hierarchies that are too rigid for Agile to be fully utilised.
- The whole society is structured in silos.
- Countless years of planning.
- The need for everything to be perfect.
- Very traditional societal structures and values.

It is not only the slow adoption process but also specific challenges like those mentioned above, which hamper the general adoption curve. Studies note some adoption challenges (Conboy et al., 2010:49; Hajjdiab & Taleb, 2011a:8-9; Gandomani et al., 2013:622; Dikert, Paasivaara & Lassenius, 2016); however, the amount of academic literature in this field is insufficient for conclusive deductions. If these Agile adoption challenges were to be understood better for the specific industries they can be found in, solutions could be found quicker and much easier (Conboy, 2009:340).

These adoption challenges are unique from company to company and from industry to industry due to the nature of service or product provided. It is clear from VersionOne (2015:6) that a significant contingent of software development also takes place in other industries such as financial services (14%) and professional services (11%). Due to this diversity, solutions to specific adoption challenges may vary according to the company, industry, and complexity of the environment. This diversity could further influence the success factors of the solution outcome.

This study focused on providing a mapping between Agile adoption challenges and possible solutions to these challenges in organisations and provided an automated means of determining the possible solutions to Agile adoption via an online adaptive questionnaire.

One of the outcomes of the dissertation was to develop a website that would automatically devise possible solutions to the Agile adoption challenges identified through the system. This system has two primary inputs:

- A database of adoption challenges and solutions developed from the outcomes of the first two key dissertation sub-objectives mentioned below.
- User selections based on the questions put forward. Questions put to the users of the system will adapt according to the selections made throughout the questionnaire.

The responses of the online participants utilising the Agile adoption online adaptive matrix (the website) in this study, was collated and the system suggested the focus areas requiring attention, as well possible solutions to these problematic areas. The suggestion

of possible solutions was not only based on the mapping of challenges to solutions but also the priority of the challenges themselves. A quantitative study in the second phase of the study determined the priority of challenges. It looked in this respect at the impact of an issue as well as the difficulty in resolving it. Dikert et al. (2016:48) suggest further study in the field of Agile adoption challenges concerning critical challenges experienced by companies, as well as which challenges are the most important. The need to identify the importance helped with the ranking of the challenges.

1.2 Problem Statement

Organisations do not currently have a rapid means of either identifying their Agile adoption challenges or finding probable and possible solutions without relying heavily on an Agile coach. Noruwana (2010:97) indicates that there is a need for developing a framework that would make it easier for organisations to adopt agile methods. This adoption framework was one of the critical objectives of resolving the problem being discussed. It is not focused on the process of adoption, but rather on mapping the challenges to possible solutions and then providing a priority for the solutions in the form of a tool.

What precisely is the problem? The onerous traditional Agile coaching approach to Agile transition costs organisations considerable sums of money. As a transformation can take a long time in larger organisations, the organisation might lose motivation for the transition if a quicker return on investment (ROI) is not realised (Noruwana, 2010:69). Organisations, therefore, require a faster method to understand their Agile adoption challenges and solutions in order to bridge these challenges.

A further problem that could arise due to the substantial involvement of an Agile coach is that specific and significant challenges or solutions may be missed due to human factors (Kitchenham & Jones, 1997:15). There are two ways that human factors can play a role in not identifying all the applicable challenges. The first scenario relates to the fact that Agile coaches typically make use of assessments and interviews (Sidky & Arthur, 2007b:37,50) to determine Agile challenges. If for some reason, the assessment does not ask all the correct questions, or the interview was not in-depth enough, a suboptimal

list of challenges will be solicited. The second scenario is where the interviewee's response is compromised by specific human factor effects (e.g. Hawthorne, learning curve, placebo or doctor effect). These effects are further explained by Kitchenham and Jones (1997:15). Furthermore, Agile adoption solutions are missed because there is no exhaustive mapping between Agile adoption challenges and possible solutions.

The following principal reasons as to why these concerns are problematic are:

1. Time wastage
2. Quality of adoption outcome
3. The cost of the transition that is impacted

Table 1.1: Problem statement items

Noruwana (2010:74) and VersionOne (2015:11) argue that suitably skilled Agile people are difficult to find, especially in South Africa, and there is subsequently simply an insufficient number of proper Agile coaches available to help teams and companies with their transition journey. Due to time constraints or lack of certified coaches, companies forge ahead with the transition with internal people who are knowledgeable or mediocre Agile practitioners, which leads to poor quality or failed implementations (Sahota, 2012:4). O'Connor and Duchonova (2014:143) indicate organisations, in general, perceive Agile coaches as being expensive. Organisations, who had previous negative experiences, mostly proceed reluctantly further with it, due to the perceived ROI (O'Connor & Duchonova, 2014:143).

The traditional approach to Agile transition is the utilisation of an Agile coach who would assist in identifying the adoption challenges manually, devise a list of possible solutions and finally work out a plan to resolve those challenges (Hajjdiab & Taleb, 2011a:6). Hajjdiab and Taleb (2011a:8) describe several case studies where the traditional approach is utilised. In these studies, the authors noted the standard processes of Agile adoption with dedicated Agile coaches. The challenges involving time, quality and cost of this approach have been discussed above.

Other manual frameworks help in the identification of the actual adoption challenges (Qumer, Henderson-Sellers & McBride, 2007:7), but the authors did not present an explicit mapping to the possible solutions, and it was also not an automated approach.

1.3 Purpose Statement

As this system, after going live, will ultimately receive online inputs from users, evaluate their answers, adapt according to input and then propose possible solutions, it inherently acts, in a limited way, as an online Agile Adoption Coach (AAC). It serves to enable Agile coaches and present them with an easily automated tool determining the most probable Agile adoption solutions for companies keen to become more Agile.

The AAC's automated solution system results can guide coaches in their decision-making to a course of action in the transition process (which would previously have been quite onerous).

Besides the distinct advantages of time saved (for both the coach and company), reduction of process complexity and cost savings to the organisation, this system's results and recommendations will not be tarnished by possible flaws in the coach's interpretations of the Agile adoption challenges in the organisation.

The paragraphs above talks to the ultimate purpose of the AAC being utilized in practice. The purpose of this dissertation, however, was to test whether the community would find this tool acceptable, accurate and useful as an instrument for understanding possible Agile adoption solutions given specific adoption challenges identified. *The above statement is also the primary hypothesis of the research.*

The prerequisite for utilizing the tool is that the person aiming to use it already understands the basic existing adoption challenges in his or her organization.

In order to formulate the AAC, key attributes have to be understood and studied in detail. These attributes are positioned as secondary research questions under section 1.4 below. They also form a significant part of the purpose of this study.

The AAC is a unique, free tool for recognised coaches to be utilised when an organisation wishes to embark on an Agile transition journey or to help with the addressing of perceived Agile adoption challenges. This study, therefore, intended to contribute towards bridging this gap by proposing an automated framework for agile adoption coaching in the software development industry.

1.4 Research Question

Primary research question: Will an automated agile adoption framework be accurate, acceptable and useful to Agile coaches, helping companies adopting Agile software development?

The emphasis in this question is whether the *framework* will be useful in helping companies to adopt Agile software development. Obviously, an Agile professional is needed to operate the framework and interpret the outcomes for the company.

The secondary research questions below do not serve to help answer the primary research question, but rather to enable and create the framework for the AAC in order that the primary research question can be answered.

a. What are the general, unique lists of Agile adoption challenges and success factors in literature? Adoption challenges and success factors are *general* in this context as it does not purport to illustrate a fully comprehensive list that cannot change. The challenges and success factors lists are *unique* in that it normalized the output of the various lists gathered from the literature.
b. What is the general priority of the various Agile adoption challenges in software development industries? The aim is to deduce the priority of implementation of the adoption challenges. The term "*general*" refers to the output of the quantitative survey on priorities and emphasize the fact that it is not an exact science.
c. What is the relationship between the various Agile adoption challenges originating areas? This question aims to discover what the different Agile adoption challenges originating areas are and if there is a particular relationship between these originating areas.

d. Are all industries involved in software development similarly impacted by Agile adoption challenges?
e. What are the other variables relevant to Agile adoption challenges in Agile software development? These “variables” relate to possibly finding unique groupings of Agile adoption challenges that can be significant for use in the AAC. The answer to this question will assist in building a comprehensive filter for the automated Agile adoption coach.
f. What is the mapping between Agile adoption challenges and recorded possible solutions to these challenges in the Agile software development space?

1.5 Research Objectives

The primary objective was to develop and test an Agile Adoption Coach (AAC) with coaches, based on the outcomes of the sub-objectives stated below. The AAC can be utilised as an assessment tool as an initial step towards Agile adoption. It will ask a series of questions to end-users, and then suggest possible solutions to their Agile challenges.

a. The first sub-objective was to identify a comprehensive list of challenges and success factors using a quantitative literature study.
b. The second sub-objective was to identify the priority of the various Agile adoption challenges using a survey. The study achieved this by considering two variables, namely the severity and difficulty to resolve a specific Agile adoption challenge.
c. The third sub-objective was to explore the relationship between the challenges’ originating areas.
d. The fourth sub-objective was to discover how Agile adoption challenges were distributed among various industries, and if it was a contributing factor using a literature study of the Agile adoption challenges.
e. The fifth sub-objective was to locate any other variables describing Agile adoption challenges using the literature study. The goal was to assess whether it can be applied to narrow the filter in the automated Agile adoption coach. There was an assumption that there might be other variables describing Agile adoption challenges which were unknown.

f. The last sub-objective was to map the adoption challenges to possible solutions comprehensively. Having this broad framework, one could find viable solutions to specific adoption challenges.

1.6 Research Methodology

The research design and methodology were broken down into three phases to cover the various research aspects of the study. The study incorporated three research areas (explained in Table 1.2 below). These phases translate to the subsequent evolvement of the study, which culminated in the AAC as the eventual product.

The research design approach was based on the “research design onion” developed by Saunders, Lewis and Thornhill (2016:124) (presented in Figure 3.1 in chapter 3). Saunders, Lewis and Thornhill (2016:124) use the onion analogy to explain how and when the individual layers of the research process needed to be addressed. This research onion has undergone several changes since the publication of Saunders, Lewis and Thornhill’s first design in their 2003 edition (compare Figure 3.1 with Figure 1.1 below). Besides various refinement changes, a whole new layer (namely methodological choice) was added.

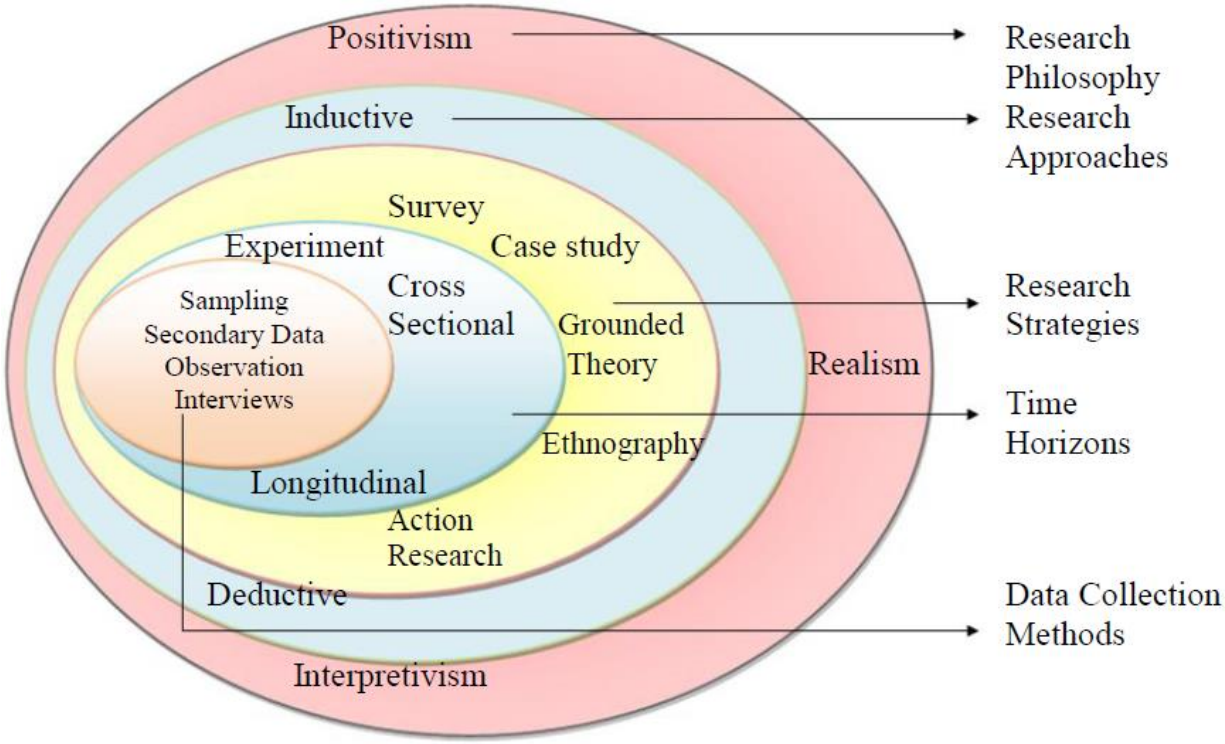


Figure 1.1: Research onion from (Saunders, Lewis & Thornhill, 2003) **– 3rd Edition**

- Phase 1 (Challenges & Solutions): It speaks to the quantitative study as part of the literature review outcome. This study helped to create a more substantial portion of the AAC database. It generated the normalised lists of challenges and possible solutions.

- Phase 2 (Challenges Prioritisation): It speaks to the quantitative study in determining the priority of the various Agile adoption challenges. This data helped to complete the AAC database. More specifically, this data is essential to the AAC so that a prioritised list of challenges could finally be recommended for any given organisation.

- Phase 3 (AAC acceptability): It speaks to the quantitative study in understanding the usefulness, usability, and acceptance by the Agile coaching community of the AAC tool. This study is the culmination of all prior work in this research.

Table 1.2: Quantitative method phase approach in research design

Table 1.3 below provides a succinct overview of the various selections made in respect to the research onion in support of this study. Each of these concepts and reasons for selection is discussed in detail in Chapter 3.

	Phase 1 (Challenges and Solutions)	**Phase 2 (Challenges Prioritisation)**	**Phase 3 (AAC acceptability)**
Philosophical approach	Positivism	Positivism	Positivism
Theory Development	Inductive	Inductive	Deductive
Methodology	Multi-method Quantitative		
Research Strategy	Survey	Survey	Survey
Method/Instrument	Literature Research	Online Questionnaire	Online Questionnaire
Time Horizon	Cross-sectional	Cross-sectional	Cross-sectional
Data Collection	Non-probability Sampling	non-probability sampling	Non-probability Sampling
Sampling Technique	Purposive	Volunteer: Snowball, Self-selection	Volunteer: Snowball, Self-selection
Sample Size	Challenges:17; Solutions: 29	30+	30+
Sampling Frame	Not available Participants: Literature	Not available Participants: Scrum Masters and Agile coaches	Not available Participants: Scrum Masters and Agile coaches
Data Type	Descriptive (Dichotomous)	Continuous Interval Data	Continuous Interval Data
Data Analysis	Descriptive	Descriptive	Descriptive

Table 1.3: Summary of Research Design Selections

For each phase, the relevant statistical analysis outputs were used based on the output of the JASP analytical tool and Microsoft Excel.

1.7 Dissertation layout

This dissertation is delineated into six chapters.

1. Introduction
2. Literature Review
3. Research methodology
4. Priority Analysis and completing AAC database

5. AAC survey data analysis and interpretation of results
6. Conclusion

Chapter 1. Introduction: This chapter provides context for the study by articulating its background, general motivation and proposed outcomes. It expands on this motivation for the study by presenting a purpose statement which clarifies how the results of the study can be used in industry.

It further clearly articulates the problem statement, the research questions and objectives, communicates a brief overview of the research method, explores the ethical considerations and finally presents a construct concerning the dissertation layout.

Chapter 2. Literature review: This chapter is introduced through a comparison between current Agile adoption frameworks and the AAC to ascertain whether a possible gap exists in the current Agile adoption approach. It continues to explore the literature concerning obtaining a unique and extensive list of Agile adoption challenges and success factors or possible solutions. It provides critical commentary in both these sections and attempts to locate a mapping between the challenges and the possible solutions. A specific gap in the Agile adoption field is articulated, and finally, the theoretical framework of the research is presented.

Chapter 3. Research methodology: This chapter describes the research design and methodology employed in the research. It delves into the various instruments used, which supports and justifies the research approach.

Chapter 4. Priority Analysis and completing AAC database: Analysis is done on the Agile challenges priorities by utilising specific techniques and methods. The output of the analysis is the last piece of the puzzle to complete the AAC database.

Chapter 5. AAC survey data analysis and interpretation of results: This chapter interprets the results of the survey by leveraging of specific analysis methods. It organises the findings in various approval categories and finally presents an outcome.

Chapter 6. Conclusion: This chapter summarises the findings from the previous chapters and asserts whether the relevant research questions are answered and objectives met.

It goes on to explore the implications of the research, discusses some limitations of the study and presents recommendations for future research.

Chapter 2: **LITERATURE REVIEW**

2.1 Introduction

Considering a core value such as “working software, over comprehensive documentation”, and a principle such as “Our highest priority is to satisfy the customer through early and continuous delivery of valuable software”, it is clear that the initial intention of the Agile manifesto was to focus explicitly on the software engineering field and how to make it more Agile (Beck et al., 2001). Despite this, the area of Agile adoption is an ever-evolving field and is still in its teenage phase, as Steve Holyer remarked in ((Denning, 2016). Denning (ibid.) compares the Agile Manifesto to the English Magna Carta or the American “Declaration of Independence” in it being a tremendously important document which could add significant value to companies; however, we need to progress as this “teenager” naturally matures. Some evidence to this maturity can already be seen in the purposeful change of “working software” to “working product” by many Agilists (Denning, 2016; Koziy, 2017) and academics (Bloomfield, 2015:5; Igbal, 2015:10).

The motivation for the use of the term “product” as part of the adopted Agile values is that Agile adoption or transformation is increasingly seen as not only useful for **software engineering** departments but also entire organisations (Song et al., 2012; Vandersluis, 2014).

Despite this tendency in the evolution of Agile in industry, the primary focus area of Agile is still in the software engineering space as supported by the Agile values, principles, practices and various methodologies which Agile underscores (Abrahamsson et al., 2003:8). These methods are being used mainly in **software engineering**, specifically **software development** (Alam & Chandra, 2014:40)**,** as a means to deliver small to huge software development projects and programmes in an Agile manner.

The discussion in the literature review focuses on the areas provided in the flow diagram below. The flow in Figure 2.1 is a logical representation of the steps to follow to generate the AAC. The need for a tool such as the AAC has been elaborated on in the “Purpose Statement” section. The purpose statement also presented the hypothesis for the acceptability of the AAC.

Each of the main areas in the flow is discussed separately to ensure a comprehensive overview of the literature has been achieved.

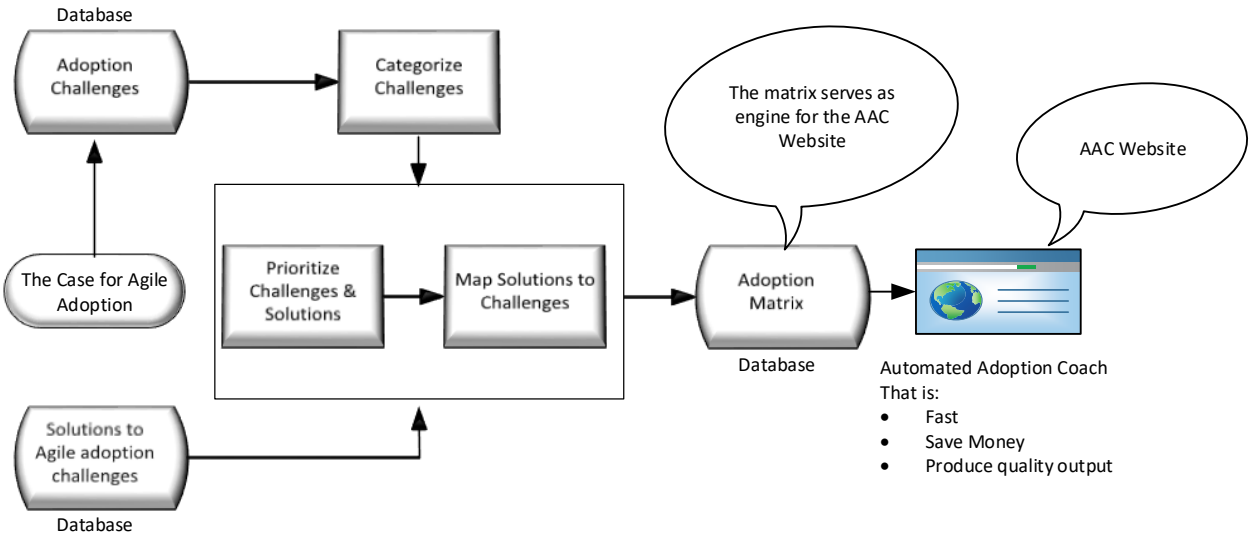


Figure 2.1: Flow diagram showing research problem concepts and their relations

As a first step, the case for Agile adoption is explored. This exploration is followed by a review of the importance of Agile adoption. The study considers various frameworks and the potential value that the AAC might add to an Agile transformation, and finally, how this study might assist in a smoother Agile transformation process.

The literature review addressed the following five of the six main sub-research questions:

a. What are the general unique lists of Agile adoption challenges and success factors in literature?

b. What is the relationship between the various Agile adoption challenges originating areas?

c. Are all industries, doing software development, similarly impacted by Agile adoption challenges?

d. What are the various other variables describing Agile adoption challenges in Agile software development? The answer to this question will assist in building a comprehensive filter for the automated Agile adoption coach.

e. What is the mapping between Agile adoption challenges and recorded possible solutions to these challenges in the Agile software development space?

Another hypothesis is relevant at this stage, which refers to the first sub research question.

If comprehensive literature research is done on Agile adoption challenges and success factors, we would be able to get to a comprehensive list of Agile adoption challenges and success factors.

As the literature review will illustrate, besides the primary research question, it is only the second sub-question: "What is the general priority of the various Agile adoption challenges in software development industries?" that will not be addressed adequately in this chapter. This sub-question is discussed in its entirety in chapter 4.

2.1.1 **Empirical versus defined process control**

Software development methods are broadly categorised into those that follow the empirical process control model versus those that support a more defined approach (Meier and Ivarsson, 2013:103). An empirical process control model asserts that knowledge comes from human experience and decision-making based on what is known and transparent in the immediate future, whereas the defined process model requires that every piece of work be wholly understood (Schwaber & Sutherland, 2017). The defined process model understanding is that given a well-defined set of inputs, the same outputs are generated every time. The defined process model has historically worked well in more traditional waterfall environments (Guckenheimer & Loje, 2012:75). Scrum and other Agile methodologies employ an iterative, incremental approach to optimise predictability and control risk.

2.1.2 **Overview of Agile versus Traditional methods**

This overview does not intend to elaborate on the various methods in-depth, but to illustrate the main differences between traditional and Agile methods, how these methods are applied and provide context for the central theme of Agile adoption in general.

The empirical approach and defined process model can be mapped to the Agile set of methods and the traditional group of methods, respectively. Some of the methods under the category of traditional methods are Waterfall, Prince2, Spiral process, V-Model and Unified Process (Mbuya, 2016:1). Agile as a philosophy (Tripp, Saltz & Turk, 2018:5466)

is acting as an umbrella-term (Alam & Chandra, 2014:39; Denning, 2015) for a number of methodologies, including Kanban, Scrum, Extreme Programming (XP), Lean, Crystal, Dynamic Systems Development Method (DSDM), Feature-Driven Development (FDD) and Scaled Agile Framework (SAFe). As part of an Agile adoption approach, traditional methods would not generally be considered part of the solution, as the nature of software (the unknowns, intangibility, customers changing their minds), would necessitate a more empirical process which lends itself to the Agile set of methods.

Each of these Agile methods follows the guidance of the Agile principles and values as a foundation, even though each of them is unique. Organisations employ these methods in light of considerations of complexity, scope, culture, development focus area (maintenance, R&D, new or enhancing of a product) and familiarity.

2.2 The case for Agile adoption

In the academic literature, there are numerous examples of Agile adoption frameworks (see Table 2.1). These frameworks differ in the complexity of implementation, role players involved, perceived success, focus on Agile parameters and Agile adoption focus (namely, whether it focuses on Agile itself, the practices or the Agile methods).

For this study, it is irrelevant what Agile adoption framework a company selects for its transformation. The AAC focuses on the initial part of any adoption framework where a typical assessment will be conducted to understand Agile maturity and what possible challenges might need to be addressed. The AAC's focus is to streamline this phase with quicker and higher quality output.

Sahota (2012:4) asserts that culture is the biggest challenge for any transformation (Adoption of Agile Practices, Agile method adoption or pure Agile adoption) and focuses predominantly on this challenge to transformation throughout his study. Agile, as a whole, is dealing with this significant shift. The impact of an Agile practice or method adoption would, therefore, be far more meaningful if Agile adoption is coping with cultural change.

It is subsequently logical that formal material addressing any flavour of Agile transformation has to presuppose an Agile adoption to be successful. If an Agile transformation is not the primary focus of a company, Agile adoption should still precede

or be run in parallel with the Agile method or practice adoption, notwithstanding the framework used.

2.2.1 **Agile adoption / maturity frameworks**

The frameworks in Table 2.1 are discussed to understand whether the AAC might be useful to any of them. This framework list contains the most common frameworks that the author could find referenced in the literature. There could be other frameworks that might be applicable to this study, but they were not found in the reference archives used.

Framework Focus	**Reference**
Accepting Agile methodologies	(Chan & Thong, 2009)
Agile transformation (focus on practices)	(Gandomani & Nafchi, 2015)
Agile deployment framework (focus on practices)	(Pikkarainen et al., 2012)
Complex Agile adoption process (focus on practices)	(Qumer & Henderson-Sellers, 2008)
Agile roll-out using a 7-dimension 5 level maturity index	(Benefield, 2010)
Team-level adoption focusing on four concepts of the Agile principles (Focus on the Agile coach)	(Kairi, 2011)
Multiple Agile adoption frameworks	(Sahota, 2012)
Agile adoption framework (SAMI and 4-Stage process)	(Sidky, 2007)
Agile transition through the identification of challenges	(Misra et al., 2006)
Agile transition with an emphasis on manager impact	(Chen, Ravichandar & Proctor, 2016)

Table 2.1: Agile adoption and maturity frameworks

Chan and Thong (2009:811) present both a theoretical and practical framework for accepting Agile methodologies in a company. Their framework starts by looking at the categories:

1. Ability (Success factors: software development methodology (SDM), self-efficacy, experience, training and external support)

2. Opportunity (Success factors: teamwork, communication, shared understanding, and arduous relationship)
3. Motivation (Success factors: career consequence, top management support, voluntariness, subjective norm, and organisational culture) related factors in a company
4. Knowledge management outcomes (Success factors: knowledge creation, -retention, and -transfer)
5. Agile methodologies (Success factors: perceived usefulness, ease of use, compatibility, result demonstrability, and maturity)

The first three categories feed into knowledge management. This knowledge management, together with the manner in which Agile methodologies are viewed, would add to an understanding of the acceptance of an Agile methodology. This framework speaks to a conceptual path to the understanding of acceptance of an Agile method and is not a framework of adoption of Agile per se. It speaks more about the readiness for innovation (Chan & Thong, 2009:810) than prescribing ways to adopt Agile.

The focus is on the various success factors which have been highlighted under the five categories above. The presumption is that if these areas are appropriately addressed, an Agile method will be accepted. The framework of Chan and Thong (2009:811) does neither make use of an assessment, nor does it refer to one being done, as part of the process. Every company is unique, yet it is the presumption that a set of success factors are inherently all that they will need to make the transition effective. Based on these identified gaps, this study proposes that the AAC could be added to the initial steps of the process, with the aim of improving rigour in the roll-out of the practical framework of Chan and Thong (2009:812).

Gandomani and Nafchi (2015:204) prescribe an empirically developed transition framework to facilitate Agile transformation. This transition framework makes use of the Scrum principles to drive the transition process. A prioritised Agile practice backlog is kept, and iterations of implementation of the top practices are initiated. After an iteration, the outcome is assessed, adjusted and then adopted. Utilising such an empirical

approach is a brilliant way to implement Agile practices within a company at their own pace. Two things, however, are not focused on sufficiently:

1. Gandomani and Nafchi (2015:207) address Agile practices adoption, yet not the principles and values of Agile per se.
2. They further do not focus on the core challenges and solutions to Agile adoption, but rather on the practices (Gandomani & Nafchi, 2015:216).

The structural characteristics (value-based, iterative, continuous gradual) all speak to an empirical approach to learning and adoption. The fact that the authors have included "assessment" as part of their principal activities also assists with continuous improvement. Utilising the AAC as part of their incremental process (in the assessment step) and even upfront would have assisted with their transformation backlog to determine not only how practices' adoption is fairing, but Agile in general.

Qumer and Henderson-Sellers (2008:1899) propose an Agile adoption framework which the authors called Agile Adoption and Improvement Model (AAIM). AAIM defines three blocks and six stages of Agile adoption:

- Prompt: Agile Infancy
- Crux: Agile Initial, Agile Realisation
- Apex: Agile Value, Agile Smart and Agile Progress

Specific practices need to be followed to reach each stage, such as quality production, the use of minimal possible resources, and keeping the process Agile. This framework utilises the Four-Dimensional Analytical Tool (4-DAT) for assessing Agile methods. AAIM is focused on some of the principles and group them in such a way that would speak to Agile maturity levels. Reaching the quantitative outcome of an Agile maturity level utilising the 4-DAT makes it vulnerable to correctness. The designation of an Agile practice on a particular stage is further subjective. In this manner, valuing people is a stage 4 practice, but communication is a stage 2 practice. One could argue that to have excellent communication; one has to value people. Considering all the steps which need to be taken, applying this model to 4-DAT could be expensive for smaller companies.

According to Qumer and Henderson-Sellers (2008:1901), the 4-DAT analytical tool is extensible with more dimensions. The problem is that the framework is already quite elaborate and complicated, and to add more complexity feels counter-intuitive. This stage in the adoption process is, however, probably the best place to add the AAC as this tool's purpose is to measure the agility of the company. It would further be hugely beneficial for a company utilising the 4-DAT to be able to identify the core challenges and solutions and how to bridge them.

Benefield (2010:2) asserts that it has been proven that to focus on seven practices/dimensions, and to ensure their fulfilment, leads to a high transition towards Agility. Besides the seven dimensions, Benefield (2010:3) allocates five levels of maturity to dimensions depending on the contemporaneous assessment. Agile coaches do assessments frequently to establish improvement of the teams concerning the dimensions.

The focused attention on these specific practices will address some critical Agile practices and some principles, but it is not a framework for Agile adoption as a whole. Such adoption takes into account a much broader set of variables. The AAC could be of value in this framework as it could identify core challenges that might inhibit individuals from adopting specific practices and subsequently provide recommended solutions on the manner in which to approach them.

Kairi's (2011:44-45) framework speaks of Agile adoption, but more to team-level coaching and adoption through its KMS model. This KMS model speaks to four categories of Agile principles and how the coach concerning the development teams measures them. It is similar to, and based on, Sidky's (2007) SAMI model. The KMS model is, however, not focused on the Agile adoption for the entire organisation.

There is further a specific focus on the involvement of an Agile coach through this journey and concerning the four concepts. The AAC could easily be incorporated as part of the KMS: both as an initial step to ascertain the scope of the challenges as well as during the rollout to measure progress on the concepts.

Sahota (2012:xii) present a framework for understanding adoption and transformation. It is, in effect, a framework consisting of other frameworks and models. These models and frameworks can be utilised depending on the context of the organisation. The various contexts include:

- The adoption of Agile practices in an inharmonious culture
- The adoption and transformation of Agile in a supportive culture
- The transformation of Agile

It is clear that in any of the frameworks utilised and suggested by Sahota, the AAC could have a place. Although the type of organisational Agile adoption (Agile practices or plain Agile) which is embarked on is irrelevant, it would always be prudent to note and understand what the current underlying challenges are and how to approach them.

Sidky (2007) made an intense study on the adoption of Agile practices in a company. He introduced a framework that consists of two parts: the first being an Agile measurement index called the Sidky Agile Measurement Index (SAMI). This index has five Agile levels that are used to measure the Agility potential of projects and organisations. It focuses on both Agile principles and practices. The second part is a 4-Stage process, which is the framework of Agile adoption. These two parts work together to align the adoption efforts of an organisation. As previously stated, the focus is not on Agile transformation but rather on Agile practices adoption.

In both the SAMI (focusing on Agility measurement) and the 4-stage process (Discontinuing Factors, Project-Level Assessment, Organisational Readiness Assessment, Reconciliation), the AAC could be applied. Significant benefits can be gained by using the AAC during any of the first three stages of the 4-stage process - especially the first stage. The outputs will be both a better measurement and provide a better understanding of the outputs of the AAC if the AAC is used before running the SAMI model.

Pikkarainen et al. (2012:678) suggest an Agile deployment framework that consists of four steps:

1. Select Agile practices.
2. Plan the deployment.
3. Execute the deployment.
4. Analyse, improve and package.

An Agile coach can apply this framework to both the organisation and a given project. With the focus on the adoption of Agile practices, the belief is if the team or organisation has gone through the framework, there would be a higher level of Agility. Pikkarainen et al. (2012:686,687) further provide a checklist of possible barriers for verification to ensure a coach is addressing all the essential obstacles and challenges. The AAC can replace this checklist with ease as it has a much more comprehensive list of challenges and probable mapped solutions.

As part of the “Select Agile Practices” phase, Pikkarainen et al. (2012:686,687) suggest that various assessments can be conducted to ascertain possible improvements. As a viable assessment tool, the AAC could be used in this stage as well.

Misra et al. (2006:27) propose a conceptual framework for the transition from a more traditional method to an Agile method. The core of the transition is the identification of the various core challenges. Based on these challenges, they describe the ideal Agile environmental change. They do not, however, mention how change needs to occur. The list of challenges is further concise, and the process to get to this ideal state is not addressed. The AAC would, however, be handy to sketch a more holistic picture of the challenges and the possible ways to overcome them. The process of how to get to the end state is not within the scope of the AAC.

Chen, Ravichandar and Proctor (2016:635) reported on the Agile transition Cisco systems went through. Their focus is on two central questions: the transition towards an Agile development process, and the identification of management practices in the new Agile environment. Chen, Ravichandar and Proctor (2016:638) focus on the role of the managers and suggest they should be seen as team coaches in addition to playing an essential part in the transition. Concerning a transition framework, it is recommended that

managers can assist with a change in the three following areas (Chen, Ravichandar & Proctor, 2016:643):

1. Performing a benefits assessment.

2. Performing a readiness assessment (which investigates leadership buy-in, task interdependence, and task-staging).

3. The most substantial multi-step area is investigating the various transition components (which includes Agile training, embedded coaching, improvements identification and an internal website to show the knowledge base of the transition - including best Agile practices).

As part of the readiness assessment step, it would be valuable to conduct the AAC as it would identify what the highest priority items are and if they are solvable or not.

In all cases of reviewing the Agile adoption and maturity frameworks above, it was shown that the AAC could have assisted in the quality and scope of the outcomes. At a minimum, it could have provided Agile coaches and management with valuable information for decision-making as to the manner in which to approach the Agile transformation further.

2.3 Literature review research strategy

As a base for this study, it is essential to have a comprehensive understanding of what the Agile challenges and success factors are. The aim was to come up with an extensive list of Agile adoption challenges and success factors using a ***survey*** on the available published literature, utilising the approach of theoretical literature research. This research would satisfy the first sub-objective.

Hakim (2000:4) defines theoretical literature research as an approach which is principally preoccupied with producing knowledge for understanding, and usually within the framework of single social science discipline. The focus area of this particular research category is specific and narrowed to the views of well-qualified Agile authors on Agile adoption challenges and success factors.

Sukamolson (2005:4) states that surveys use scientific sampling to measure the characteristics of a population with statistical precision. It typically seeks answers through questions such as “How many …?” and “How often …?”. Sukamolson (ibid.) mentions that a survey provides estimates from a sample group which can be related to an entire populace to a certain degree of certainty. Random sampling is key to the success of a survey where the population is too large to observe directly (Sukamolson, 2005; Babbie & Mouton, 2012).

A quantitative study of the literature provides a much broader and deeper outcome in the context of Agile challenges and success factors than, for example, a survey questionnaire research approach. A proposed sample group of Agile coaches need to be quite substantive to cover all Agile adoption challenges and success factors as found in literature, for example. There are firstly merely not enough coaches, and, secondly, those who would respond would most probably not be willing to take the time to think of a comprehensive list of challenges and success factors. The answers provided would furthermore mostly be subjective and not necessarily a complete and descriptive list of suggested challenges and success factors. The respondents would typically not be willing or inclined to fill in entire pages of possible adoption challenges. These answers will further usually focus on the most top-of-mind challenges of the respondent and could then easily leave out lesser-known challenges. Case studies would similarly merely provide a very narrow focus on the problem statement as the literature review provides access to numerous existing case studies. Pursuing another would not fulfil the objectives.

To satisfy the aim, it would be more prudent to survey the available literature on Agile adoption challenges and success factors. These texts usually describe in much depth the challenges (even though one has to be careful as many authors do confuse general Agile adoption challenges with adoption challenges of one of the Agile methods). Still, the scope of input from these studies is of significantly higher quality and depth than what would possibly be ascertained from other means as these studies in themselves usually already make use of large input samples.

2.4 Agile adoption challenges

2.4.1 Introduction

A challenge in the field of Agile adoption research is locating literature that focuses explicitly on Agile adoption challenges in organisations. There are numerous variances on Agile adoption studies. There are only a few studies that clearly and directly state the challenges.

The approach in this section is to devise a normalised list of challenges. This list should illustrate the following:

1. The industry in which the challenge can be found. If it is not explicitly stated, it is assumed that it can occur in any industry where software development happens.
2. The related Agile solutions to the challenges (if mentioned by the literature).
3. The root area where the challenge is originating from, as discussed by Gandomani et al. (2013:622) and Nerur, Mahapatra and Mangalaraj (2005:76). The author introduced two more originating factors: developer and Agile rollout challenges (further explained in Table 2.21 below).
4. The types or categories of challenges, as mentioned by Dikert, Paasivaara and Lassenius (2016).
5. The difficulty in resolving a challenge (if provided), as well as the impact if not addressed (i.e. the severity). These numbers were worked back to a value out of five.

When any duplicates were found, the combination of findings of the above criteria was applied to the unique challenge.

The referenced material was selected in such a way that it would provide an extensive list of challenges from which the AAC user can choose possible challenges.

This referenced material was obtained in two ways: a Google Scholar search was conducted on the 17th of November, 2017, on the search terms “agile adoption” and challenges. The top 40 relevant results were scrutinised in order to ascertain to what depth the authors delve in their studies concerning the explanation of challenges, number

of challenges and its relevance to Agile adoption. Eight out of the 40 studies were selected to include in this study, and are listed according to the Google scholar ranking. The other thirty-two results either made trivial remarks towards challenges or positioned a rudimentary list of challenges that were already contained in other studies without much depth. Some studies just relisted the challenges as were found in the studies in Table 2.2 below.

1. (Marchenko & Abrahamsson, 2008)
2. (Conboy et al., 2010)
3. (Hajjdiab & Taleb, 2011b)
4. (Vijayasarathy & Turk, 2008)
5. (Rodríguez et al., 2012)
6. (Hajjdiab & Taleb, 2011a)
7. (Srinivasan & Lundqvist, 2009)
8. (Gandomani et al., 2013)

Table 2.2: Selected studies from top 40 Google Scholar relevant results

The second resource list of nine studies, mentioned in Table 2.3, is a random selection of studies providing a broad overview of challenges and possible solutions. The studies included in this list had to fulfil the following criteria:

1. It has to have specific sections dedicated to adoption challenges.
2. It must address challenges in the context of Agile adoption.
3. It should not only apply to a particular method.
4. It should include a unique perspective to adoption challenges.

The selected main database for this study, Google Scholar, returned 106 000 results for the above search parameters, indicating that these terms are used in the result-set, but not necessarily suggesting that there are 106 000 research studies focussing on agile adoption challenges. Despite this, published literature has no study that ascertains a much broader scope of adoption challenges with a mapped list of possible solutions.

1. (Dikert et al., 2016)
2. (Nerur et al., 2005)
3. (Noruwana & Tanner, 2012)
4. (VersionOne, 2016)
5. (Pitkänen, 2015)
6. (Mishra & Mishra, 2011)
7. (Boehm & Turner, 2005)
8. (Scrumology, 2012)
9. (Almeida, 2017)

Table 2.3: Random studies on Agile adoption challenges

An unnormalised list of 236 challenges was extracted from the sample list of challenges. This unnormalised list should prove satisfactory to obtain a normalised list of challenges that represent at least 90% of possible general Agile adoption challenges found in the software development industry.

This section does not intend to explore the challenges itself (such as the more profound consequences of it on an organisation, for example) but merely to document a comprehensive normalised list of challenges which is used as a primary input to the AAC as can be seen in Table 2.24.

2.4.2 Challenges

2.4.2.1 Conboy et al.'s (2010) Agile adoption challenges

Conboy et al. (2010:2) focus on the people-related challenges while the Agile transformation occurs. Conboy et al. (2010:3-10) highlight nine challenges and 27 recommendations and solutions to counter these challenges. The authors argue that the challenges are caused due to the transition process itself, while it is clear from studying the challenges that it is not the *process* that is the cause, but rather the challenge of adopting Agile that is surfacing the challenges (Conboy et al., 2010:2). This list of challenges contains those faced during Agile adoption, and which is one of the aspects that are of interest to this study. The challenges mentioned are:

1. Developer fear caused by the transparency of skill deficiencies.

2. The need for developers to be a "master of all trades".
3. Increased reliance on social skills.
4. A lack of business knowledge among developers.
5. The need to understand and learn the values and principles of agile, not just the practices.
6. Lack of developer motivation to use agile methods.
7. The implications of devolved decision-making.
8. The need for Agile-compliant performance evaluation.
9. Lack of agile-specific recruitment policies and suitably trained IT graduates.

Table 2.4: Summarised challenges mentioned in Conboy et al. (2010)

2.4.2.2 Gandomani et al.'s (2013) Agile adoption challenges

Gandomani et al. (2013:620-623) focus on Agile challenges which can be resolved by company managers. They start with the premise that most companies interested in Agile have many years of experience in traditional methods. The four core barriers to Agile adoption, which they identify are subsequently the result of this premise (Gandomani et al., 2013:622-623). The barriers are grouped in root areas where challenges are originating from. This means each of these areas can have multiple challenges. These include organisation and management, people, process, and technology and tools. Even though the challenges are not all listed, they do mention a few challenges under each category and also suggested solutions to some of them.

Very little is mentioned regarding people-related challenges. The authors address the need for collaboration, the need for the customer to be part of the team and the resistance faced as a result of that. They lastly illustrate that developers might resist critical practices like pair programming. This section is summarised by indicating that a lack of coaching, training and mentoring are critical reasons as to why people-related challenges are not resolved adequately (Gandomani et al., 2013:623). The research by Gandomani et al. (2013) is mostly an extension of the material of Nerur, Mahapatra and Mangalaraj (2005). They, in turn, focus on a few additional challenges under the main headings, and they have also suggested some solutions to the challenges.

For this study, all challenges are listed under (1) organisation, (2) management, (3) people, (4) process, (5) technology and tools and additionally (6) Agile roll-out challenges and (7) developer challenges. Agile roll-out challenges occur when the challenge has arisen due to an improper roll-out of Agile. Examples of such challenges include insufficient training, coaching and misunderstanding of Agile concepts. Benefield (2010:1) mentions that Agile implementations struggle to succeed in large operations, with this being due to the improper preparation of the scope of the roll-out.

Developer challenges are separated under general people-related challenges in this study to ensure a more specific root area is defined for the challenge's origin. If a people-related challenge is not particular to developers, it will fall under general people-related challenges, for example. Otherwise, it will fall under developer challenges.

Challenges	**Suggested Solutions**
Organisation and Management	
1. The organisational tendency towards command and control. 2. The manager has the role of planner and controller. 3. Managers struggle to relent previous authority and role. 4. Over-generation of documentation in traditional environments. 5. Challenge of communication in distributed teams. **6.** Cultural differences in multi-international teams.	1. Teach managers to a style of leadership and collaboration. 2. Guide managers to follow a director and coordinator model. 3. Changing mindset takes time and managers should be continually mentored and coached. 4. Documentation should be just enough. Define appropriate knowledge management strategy and distribution of knowledge in different levels of the organisation. 5. Nothing said 6. Nothing said
People	
1. Resistance to the customer or the client to be part of the team. **2.** Resistance to social-orientated Agile practices like Pair programming (XP).	1. Coaches should patiently coach the team on the value of the customer in the team 2. Select appropriate personnel and provide the necessary training, mentoring and

	creating a set of work practices that promote process excellence.
Process	
1. Changing the process model from traditional to Agile. **2.** Choosing an appropriate Agile method is hard. (De Haaff, 2017)	2. Nothing said 3. Nothing said
Technology and tools	
1. Using non-flexible tools and hardware is a barrier to moving to agile. **2.** Tools for multi-site organisations is a problem.	1. Companies should use tools that can supply incremental evolution, continuous integration, re-working, version management and support of other agile practices. 2. Invest in tools that support distributed agile teams.

Table 2.5: Categorised and summarised challenges and solutions in Gandomani et al. (2013)

Gandomani et al. (2013:622) indicate that managers should pay particular attention in assigning experienced and professional coaches in teams (also see Table 2.5). On four of the challenges, nothing was mentioned with respect to possible solutions. One has to assert that with the emphasis of coaches in teams that they will have to mitigate those challenges by way of coaching and mentoring.

Jurgen Appelo mentions in the foreword of Sahota (2012) that people usually do not struggle so much with the adoption of Agile practices, but rather with the transformation to the Agile mindset as many organisational cultures actively resist it. A number of studies (Qumer, Henderson-Sellers & McBride, 2007; Wang, Conboy & Pikkarainen, 2012; Conboy, 2009; Qumer & Henderson-Sellers, 2008; Sidky, 2007; Sidky & Arthur, 2007; etc.) focus on adoption challenges of the Agile or Scrum practices, and not necessarily on the Agile mindset itself. As there is not always a clear distinction between the two, one needs to be careful as to how to interpret the study, as Sahota (2012:IV) indicates there

is a difference in the effort of adoption. Soundararajan and Arthur (2011) illustrate that the “goodness” of Agile can be achieved through adhering to the objectives, principles, and practices (OPP) framework. OPP stresses that practices cannot stand on its own; values and principles need to come first.

2.4.2.3 Nerur, Mahapatra and Mangalaraj’s (2005) Agile adoption challenges

Nerur, Mahapatra and Mangalaraj (2005:74) focus on the challenges of Agile methodology adoption and not Agile as a unique philosophy. Again; the focus on the methodology of adoption versus that of the mindset does pose a challenge to discern whether it could be applied to Agile adoption per se. The authors devise 14 challenges residing under four main categories (see Table 2.6), which Gandomani et al. (2013:622) leveraged. The intention of the work of Nerur, Mahapatra and Mangalaraj (2005:78) is not to create a set of real solutions, but rather to raise a warning to prospective companies that are thinking of adopting Agile. This warning is to do a proper internal assessment before embarking on an Agile adoption journey (Nerur, Mahapatra & Mangalaraj, 2005:78).

Management and organisational
1. Organisational Culture 2. Management Style 3. Organisational Form 4. Management of Software Development Knowledge 5. Reward Systems
People
1. Working effectively in a team 2. High level of competence 3. Customer relationships—commitment, knowledge, proximity, trust, respect
Process
1. Change from process-centric to a feature-driven, people-centric approach 2. Short, iterative, test-driven development that emphasises adaptability

3. Managing large, scalable projects 4. Selecting an appropriate agile method
Technology (Tools and Techniques)
1. Appropriateness of existing technology and tools 2. New skillsets—refactoring, configuration management, JUnits

Table 2.6: Categorised and summarised challenges in Nerur, Mahapatra & Mangalaraj (2005)

There are numerous examples of published literature with its core focus not on the Agile adoption challenges while exploring it to an extent.

2.4.2.4 Vijayasarathy and Turk (2008) Agile adoption challenges

Vijayasarathy and Turk (2008:5) speak about the experiences of Agile early adopters and briefly mentions seven Agile adoption challenges they have faced. The authors do not attempt to identify the related solutions to the challenges. The challenges are:

• Organisational Resistance • Management Apathy • Inadequate Training • Lack of Peer Support	• Lack of Formal Guidelines • Minimal Rewards for Using Agile Techniques • Increased Risk of Project Failure

Table 2.7: Summarised challenges observed in Vijayasarathy & Turk (2008)

2.4.2.5 Noruwana and Tanner's (2012) Agile adoption challenges

Noruwana and Tanner (2012:43) focus on the processes leading up to Agile adoption without going into detail about the challenges. Noruwana and Tanner (2012:43) mention six challenges which relate more specifically to Scrum (see Table 2.8); however, it can be applied to Agile in general as well.

Noruwana and Tanner (2012:54) note that stakeholders need to have a shared idea of both the challenges and solutions, as well as how they relate to one another. The authors do not, however, attempt to find a mapping to the challenges they have mentioned.

1. Culture (Difficult to change the existing culture) 2. Lack of Structured Approach (Makes the process of adopting agile systems difficult and exposes it to failure.)

3. Assigning of New Roles (Team members find it difficult to change the roles they are used to.) 4. Slow buy-in (Resistance to change) 5. Developers Opposed to Pair Programming (Developers prefer to be private about their knowledge.) 6. Resistance to Team Evaluation (Individuals want to be recognised as individuals for their contributions.)

Table 2.8: Summarised challenges in Noruwana & Tanner (2012)

2.4.2.6 VersionOne's (2015) Agile adoption challenges

VersionOne (2015) states that key barriers to further adoption usually reside around culture, including the ability to change, general resistance to change and management support. Company culture is given as the main reason for project failure. Once the above challenges have been overcome, the limiting factor most often cited is the availability of personnel with the necessary Agile experience.

VersionOne (2016) provides a relevant summary of the Agile adoption challenges the respondents have mentioned they have experienced in their respective companies (see Table 2.9). No mention is made of any solutions to these challenges. The authors indicate various industries that the respondents came from, but do not show whether it was challenge dependent. They list the following challenges in order of most commonly found:

1. Company philosophy or culture at odds with core agile values. 2. Lack of experience with agile methods. 3. Lack of management support. 4. General organisation resistance to change. 5. Lack of business/customer/product owner.	6. Insufficient training. 7. Pervasiveness of traditional development. 8. Inconsistent agile practices and process. 9. Fragmented tooling, data, and measurements. 10. Ineffective collaboration. 11. Regulatory compliance and governance.

Table 2.9: Summarised challenges in VersionOne (2016)

2.4.2.7 Hajjdiab and Taleb's (2011a) Agile adoption challenges

In Hajjdiab and Taleb's (2011a:8-9) case study, eight challenges are mentioned which surfaced in a UAE government agency Agile development adoption effort (see Table 2.10). They compare these challenges with challenges cited by six other literature publications, namely: Srinivasan and Lundqvist (2009), Cohn and Ford (2003), Lindvall et al. (2004), Conboy et al. (2010), Cohn (2009) and Hunt (2006). They do not, however, discuss any challenges mentioned by Cohn (2009) and Hunt (2006). They merely state that challenge 6 (upper management concerns) is also present in the challenges mentioned in these texts.

1. Missing the Agile Master Role (The need for an Agile coach was recognised, but there was no budget for this individual.)
2. The overzealous teams
3. The Absent of a Pilot Project
4. Scrum Implementation (They found the mindset shift quite hard to adapt to.)
5. Current Work Pressure (The Agile project was not the only project they were working on)
6. Upper Management Concerns
7. Governmental bureaucratic System (Lots of sign-off and governance steps hindered speed.)
8. Documentation requirements (Lots of documentation provided a feeling of safety before.)

Table 2.10: Summarised challenges with an explanation in Hajjdiab and Taleb (2011a)

The comparison matrix is not a comprehensive overview of challenges in general, and therefore, it is not surprising that challenges 5, 7 and eight do not get any matches from the six works they selected. As an example of non-inclusiveness; Gandomani et al. (2013:622) speak directly to challenge eight concerning documentation requirements, and that in Agile projects, documentation should be kept to a minimum.

2.4.2.8 Hajjdiab and Taleb's (2011b) Agile challenges & Solutions

In the IEEE conference proceedings, Hajjdiab and Taleb (2011b) mention the same challenges as in Hajjdiab and Taleb (2011a), but they also continue to suggest specific solutions to individual challenges (see Table 2.11).

Chal. ID	Suggested solution
1	In agile software engineering literature, the Agile Master has a critical role to play to ensure successful outcomes. The company needs to hire an Agile Master for at least six months to increases the chance of succeeding in the agile adopting process.
2	Careful planning and discipline of agile adoption are required. A team must also expect a slight decrease in productivity when first adopting agile while the teams learn the new implementation techniques.
3	A pilot project must be selected when first adopting agile, and it should be near the middle of what is the average for an organisation, small enough to be done by one team and should not be critical to the organisation
4	To help with the implementation, the Agile master role must be provided to guide teams during the first six months of Agile implementation, hire a senior developer with experience in Agile and Scrum and increase the number of agile courses training and workshop that provide more details of how to implement Scrum and Agile.
5	The adoption of agile should be planned in a time that has the minimum work pressure, for instance, a company should invest more time by not accepting new projects for six months and investing this time only for agile adoption.
6	Upper management approval is fundamental to support any significant change in the processes of any department; without their approval, the success of the adoption is doubted. To acquire the upper management support a presentation or a meeting might be conducted to demonstrate the new agile development and explain the benefits of changing to a new method.
7	One of the valuable benefits of agile development is its ability to be customised based on the culture and the environment of the organisation it is adopted in. These changes should not compromise its beneficiary and efficiency. The

	governmental systems should be revised and enhanced to what is best for the governmental organisation.
8	What agile development offers is a new way of comprehensive documentation that takes less time and effort. This is another issue that needs to be supported by the upper management and agreed upon with the customers from the beginning of the project.

Table 2.11: Highlighted, summarised solutions to challenges in Hajjdiab and Taleb (2011b)

2.4.2.9 Srinivasan and Lundqvist's (2009) Agile adoption challenges

Srinivasan and Lundqvist (2009:1415) mention four critical challenges (see Table 2.12) in Scrum adoption at “GameDevCo”. Although the study is on Scrum adoption, the challenges are equally relevant in typical Agile adoption.

1. *Requirements not coherent.* There is not proper traceability between the user stories and the actual implementation of the product. The critical problem is that they are unsure of what is enough documentation. Adding to the lack of requirements, quality is the fact that there was not a proper tool for requirements management.
2. *Scrum Implementation.* The core reason for this being a challenge is that there was not sufficient training for current and new team members to be on the same page concerning Agile and Scrum.
3. *Organisational Learning*. Ineffective sprint review and retrospectives due to immature Scrum Masters and unhealthy focus only on process.
4. *Verification and Validation*. The two key things that were highlighted under this point were the fact that that there was not one specific system owner and secondly that quality assurance analysts were not part of the teams, and that contributed to a lower quality outcome for verification and validation.

Table 2.12: Summarised challenges from Srinivasan & Lundqvist (2009) **with explanations.**

No specific solutions are suggested for the various challenges mentioned above.

2.4.2.10 Dikert, Paasivaara and Lassenius's (2016) challenges & success factors

The seminal study of Dikert, Paasivaara and Lassenius (2016) not only identifies 29 success factors in 11 categories but further reports 35 Agile adoption challenges in 8 categories (see Table 2.13). They do not, unfortunately, attempt to implement a direct mapping between the challenges and success factors they mention.

Challenges	Challenge Category
Misunderstanding agile concepts	Agile difficult to implement
Agile customised poorly	Agile difficult to implement
Reverting to the old way of working	Agile difficult to implement
Excessive enthusiasm	Agile difficult to implement
Lack of guidance from the literature	Agile difficult to implement
Scepticism towards the new way of working	Change resistance
Top-down mandate creates resistance	Change resistance
Management unwilling to change	Change resistance
General resistance to change	
Achieving technical consistency and integration	Coordination challenges in the multi-team environment
Interfacing between teams difficult	
Autonomous team model challenging	
Global distribution challenges	
Using old and new approaches side by side	Different approaches emerge in a multi-team env.
Interpretation of agile differs between teams	
Middle managers' role in agile unclear	Hierarchical management and Org. boundaries
Keeping the old bureaucracy	
Internal silos kept	
Management in waterfall mode	
Other functions unwilling to change	Integrating non-development functions
Rewarding model, not teamwork centric	

Challenges	Challenge Category
Challenges in adjusting product launch activities	
Challenges in adjusting to the incremental delivery pace	
Lack of coaching	Lack of investment
Lack of training	
Too high workload	
Challenges in rearranging physical spaces	
Old commitments kept	
Lack of automated testing	Quality assurance challenges
Accommodating non-functional testing	
Requirements ambiguity affects QA	
The gap between long and short-term planning	Requirements engineering challenges
Creating and estimating user stories hard	
Requirement refinement challenging	
High-level requirements management is mostly missing in Agile. (Ramesh, Cao & Baskerville, 2010)	

Table 2.13: Summarised challenges with respective categories (Dikert, Paasivaara & Lassenius, 2016)

2.4.2.11 Rodríguez et al.'s (2012) Agile adoption challenges

Rodríguez et al.'s (2012) excellent survey on Agile and Lean adoption spans more than 200 organisations and 600 individuals in the Finnish context (see Table 2.14). The study provides excellent insight into a general state of Agile in Finland. Challenges are listed showing the related impact or severity weighting. These weightings are deduced using the 5-point Likert-scale as used in the survey. They do not recommend any specific solutions to these challenges or any particular industries.

1. Top management commitment 2. Customer/supplier collaboration 3. Cultural change 4. Measuring Agile success 5. Resistance to change 6. Defining business value 7. Need for specialised skills 8. Tailoring Agile/Lean practices 9. Lack of formal guidelines 10. Scalability of Agile / Lean practices	11. Inadequate documentation 12. Synchronising activities 13. Loss of management control 14. Lack of big design upfront 15. Fixed priced contracts 16. Steep learning curve 17. The inappropriateness of existing tools 18. Achieving Flow 19. Decreased predictability 20. Inadequate training

Table 2.14: Summarised challenges from Rodríguez et al. (2012)

2.4.2.12 Pitkänen's (2015) Agile adoption challenges

Pitkänen (2015:16) summarises ten Agile adoption challenges (see Table 2.15). What makes the challenges listed noteworthy is that they were firstly retrieved from published case studies and secondly only registered if they could be corroborated with at least three other case studies that referenced those particular challenges. It is further unique in that the identified challenges are specific to Agile adoption in large-scale environments.

1. Integrating other functions of the company 2. Including user-centred design processes into scaled agile 3. Requirements management with large/complex products 4. Top-down approach creating resistance towards change 5. Not understanding the agile values behind practices 6. Inter-team communication, team coordination, dependency management 7. Teams too focused on own goals instead of the bigger picture 8. Diverged agile models between teams 9. Lack of literature guidance on scaling agile practices 10. Incorporating quality assurance and performance testing into the scaled agile process

Table 2.15: Summarised challenges from Pitkänen (2015)

Pitkänen (2015:24) delves into success factors directly after exploring the challenges. The author does not, unfortunately, attempt to map the success factors to the challenges mentioned.

2.4.2.13 Mishra and Mishra's (2011) Agile adoption challenges

Mishra and Mishra (2011:551) address challenges while adopting Agile methods in large projects (see Table 2.16). These challenges can further be applied to Agile adoption in companies with multiple teams.

1. Challenges regarding realising continuous testing
2. Increased maintenance effort with an increase in the number of releases
3. Management overhead due to the need for coordination between teams
4. Detailed dependencies are not discovered on a detailed level due to a lack of focus on design
5. Lengthy requirements engineering duration, due to complex decision processes in requirements engineering
6. Requirements priority lists are hard to create and maintain
7. Waiting times in the process, specifically in design waiting for requirements
8. Reduction of test coverage due to a shortage of projects and lack of independent testing
9. Increased configuration management effort

Table 2.16: Summarised challenges from Mishra and Mishra (2011)

2.4.2.14 Boehm and Turner's (2005:30) Agile adoption challenges

Boehm and Turner (2005:30) focus on the challenges that managers are faced with in trying to move to Agile methods in more traditional organisations. They group the various challenges into three related categories and provide multiple suggestions for each category. The authors do not, unfortunately, offer a one-to-one mapping on the challenges to solutions, even though they provide general recommendations for each of the three categories.

Below in Table 2.17, Boehm and Turner (2005) show the manner in which they structure these challenges:

Development process conflicts
1. Variability (Interdependency management between various systems) 2. Different life cycles (Variance on speed od delivery between Agile and traditional) 3. Legacy systems (Does not integrate with Agile methods easily) 4. Requirements (Traditional and Agile Requirements processes at odds with one another)
Business process conflicts
5. Human Resources (time-keeping, reward systems, etc. Agile deals differently with this) 6. Progress measurement (Agile's rapid pace needs different techniques) 7. Process standard ratings (Concern about the impact on ISO and CMMI levels)
People conflicts
8. Management attitudes (How management perceive employees and their roles, differ considerably between Agile and traditional) 9. Logistical issues (Agile has a strong focus on co-location) 10. Handling successful pilots (splitting teams and removing critical resources after the pilot) 11. Lack of change management during the adoption process

Table 2.17: Categorised and summarised challenges (Boehm & Turner, 2005) **with explanations**

2.4.2.15 Scrumology's (2012) Agile adoption challenges

Scrumology (2012) presents a white-paper on Agile adoption challenges. They indicate these challenges to be the core challenges to Agile adoption that may have to be dealt with. This white-paper is very beneficial in that it describes the various proposed solutions to the challenges concisely. The challenges and corresponding solutions are summarised below:

	Challenges	Solutions

1	Team members are overly specialised, which requires work to be handed off several times.	To minimise hand-offs, encourage pairing at all stages and ensure that the necessary handoffs are informal. Encouraging the team to tackle small chunks of work also helps because it forces them to communicate more frequently.
2	Lack of ownership by the team.	Let the teamwork directly with the customer
		Encourage team members to own and solve their problems instead of letting organisations control technical conversations.
		Solicit input from different members of the team.
3	Some team members refuse to interact with the team.	Every member of the team needs to contribute, and if one or two members of the team are putting themselves above the team, this needs to be quickly addressed by management. Often, this results in the quick removal of the individual(s).
4	It can be difficult to convince management of the need for a new development approach.	The first step in educating an organisation about Agile development is speaking to the team or individual members of the team. However, you also need to consider a wider audience, including functional managers; the PMO and HR. Failure to address this wider audience can hobble the transition to Agile.
		Management requires education, coaxing and convincing to recognise that there is a better way to develop software.
		When working with management, pay attention to both Agile metrics (and the lack, thereof), and adaptive planning over predictive planning. The Agile approach to both of these is counter-intuitive for most classically-trained managers and requires constant reinforcing.
5	Senior management is giving mixed signals regarding their support for Agile.	Agile teams must frequently communicate with senior management to clearly articulate their successes and challenges, and they should share their experiences with the broader organisation, so their achievements are acknowledged.
6	No single Product Owner can be identified.	In the situation where a single customer cannot be identified, pinpoint the project sponsor, who is the sole person who ultimately approves the funding of the project. By working with the sponsor, and clearly articulating the need for a single business representative on the project team, this situation can usually be quickly resolved.
		In the situation where several different groups have an equal interest in the success of a project, the team still needs a single representative who is willing to work with each of the different

		groups and prioritise accordingly. Again, work with the project sponsor to provide the path to a solution.
7	Management wants to combine elements of RUP and Agile.	Educate management and business partners on a single approach and the value of it.
8	The Scrum Master refuses to protect the team.	If the Scrum Master is ineffective at protecting his team, this person needs to be replaced. To find a new Scrum Master, request volunteers or ask the team whom they would like to represent them.
9	You do not have a reliable build system and processes.	Implementing Test-Driven Development (TDD) and Continuous Integration (CI) are reasonable first steps. The tools for both of these practices are standard and (often) free.
10	QA issues are not addressed.	Motivate, educate and train teams in the use of Test-driven development (TDD), Pair programming, continuous integration (CI) and refactoring as starters.
11	External parties have mandated ineffective tools.	The Scrum Master and project team need to address the issue with senior management. They need to explain why specific tools are relevant and need to be supported, and they need to make management understand that having the right tools can make the difference between a good enough product and a high-quality product.
12	Friction can exist between teams.	The standard approach is to prioritise dependant functionality early and to code to an agreed-upon the interface, but this is overly simplistic. Instead, teams need to recognise that there is a dynamic relationship between dependant teams that need to be actively and continuously improved.
13	The way the company rewards employees, conflict with the “values” of Agile development.	Long-term solutions are dependent on rewarding teamwork and breadth of understanding. While there is no single or easy solution, some ideas include having team goals and bonuses, eliminating individual performance reviews, and allowing teams the opportunity to make mistakes.

Table 2.18: Summarised challenges with solutions (Scrumology, 2009)

2.4.2.16 Marchenko and Abrahamsson's (2008) Agile adoption challenges

In the case study of Marchenko and Abrahamsson (2008:15), the focus is very specifically on the adoption of Scrum in a multi-team environment. The challenges observed are mostly consistent with the challenges found in method-agnostic Agile adoption. Those

challenges which are unique to Scrum, and which cannot be related to Agile adoption, are not reflected in the normalised list of challenges listed in Table 2.24 further below. The authors state that the challenges they observed could apply to any industry. They break the ten observed challenges down concerning specific Agile / Scrum practices, roles, and where they found it. The narrative on the challenges is compelling; however, no clear solutions are suggested for the various challenges. The categories and challenges are summarised below:

Category	**Challenge**
Scrum Master	1. Placing an overemphasis on the Scrum process and practices.
	2. The Scrum Master is caring only about the individuals and interactions (and ignoring the process).
Product Backlog	3. A lack of clear management expectations and actions.
	4. Too much maintenance and bug fixing undermining the team productivity and morale.
Scrum Teams	5. Fitting Scrum and short iterations into research-intensive teamwork.
	6. Over-specialism undermining collaboration.
	7. Over-individualism.
Sprint Planning Meeting	8. Committing to too much.
	9. Difficulty in tracking progress and in using the results of the tracking.
	10. Management is interfering too much.

Table 2.19: Summarised categories and challenges from Marchenko and Abrahamsson (2008)

2.4.2.17 Almeida's (2017) Agile adoption challenges

Almeida (2017) presents a study on migration challenges from traditional / Waterfall to more Agile-based environments. In his literature review, he categorises the most cited studies on this subject and selects the six most cited studies as the primary input from which to determine the list of challenges. Similar to the studies of Gandomani et al. (2013) and Nerur, Mahapatra and Mangalaraj (2005), Almeida groups challenges under the headings of organisation and management, people, process and tools. Almeida (2017:41)

asserts that organisation and management, followed by people-related challenges, is the most prolific. Process and tools challenges follow this.

Almeida (2017) draws various mind-maps for the different categories throughout his paper, with additional subcategories to describe the challenges in more detail. His conclusion includes 49 challenges under these categories. The table below summarises these challenges. The challenges have been merged with the second level categories for ease of reading. While Almeida (2017:39) mentions that mitigations for the challenges have been addressed, other significant references to support this argument could not be found; hence mitigations for the challenges have not been included in this study.

Organisation and Management	**Process**
Legislation: Rules and procedures	Team practices and roles: Leadership
Legislation: External audits	Team practices and roles: Share of work
Culture: Organisation behaviour	Requirements: Identification
Culture: Knowledge management	Requirements: Missing
Internal silos: Departmental	Requirements: Conflicting
Internal silos: Ad-hoc relationships	Non-development functions: Identification
Change resistance: Individual resistance	Non-development functions: Emphasis
Change resistance: Group resistance	Documentation: Collaborative approach
Change resistance: Organisational resistance	Cross-team dependencies: Heterogeneity of teams
People	Cross-team dependencies: Full-stack feature teams
Personnel education: Knowledge of agile	Reporting and tracking: Metrics
Experience and commitment: Previous projects	Reporting and tracking: Estimation
Experience and commitment: Adaptability	End-to-end quality: Tests management
Stakeholder involvement: Collaborative work	End-to-end quality: Refactoring
Stakeholder involvement: Customer follow-up	End-to-end quality: Reviews and inspections
Location: Team members	End-to-end quality: Standards and guidelines
Location: Stakeholders	Risk management: Identification
Training: Availability of courses	Risk management: Prioritisation
Training: Time consumption	Risk management: Management and control
Customer needs: Identification	Scaling: Velocity

Customer needs: Changes	Scaling: Coordination
Tools	Scaling: Quality
The complexity of SW architecture: Global vision The complexity of SW architecture: Later integration System integration: Vendors System integration: Multiple product owners Project assessment: Acceptance criteria Project assessment: Continuous integration Issue tracking: Mapping	

Table 2.20: Summarised categories and challenges found in Almeida (2017)

2.4.3 **Discussion**

The normalised and unique list of challenges can be found in Table 2.24. This list was finalised after conducting a comparison of challenges between the various referenced authors.

2.4.3.1 Originating areas of challenges

As mentioned in 2.4.1, two additional originating areas were added by the author (as can be seen in Table 2.21 below):

ID	Origin	Reference
1	Developer Challenges	Added by Author
2	Management Challenges	(Gandomani et al., 2013)
3	Agile Rollout Challenges	Added by Author
4	Organisational Challenges	(Gandomani et al., 2013)
5	General People Challenges	(Gandomani et al., 2013)
6	Process Related Challenges	(Gandomani et al., 2013)
7	Technology and Tools Related Challenges	(Gandomani et al., 2013)

Table 2.21: Originating areas of challenges

In the normalised list of challenges, particular challenges are identified which are unique to developers and should not be categorised under general people-related challenges. This specific emphasis will otherwise be lost. As developers are the key group creating software, it is prudent to note specific originating challenges from this group.

“Agile roll-out challenges” is another, and is the most significant originating area of all Agile adoption challenges (as can be seen in Figure 2.2.). General people-related challenges follow this. Thirty-five out of the 97 challenges are due to roll-out challenges. It would appear as if many companies and individuals are keen on an Agile transformation, but are hampered by an ineffective roll-out procedure. Scrutinising these types of challenges in Table 2.24, it is clear that the cause of this challenge is a lack of proper Agile coach guidance for such companies. This lack of guidance is a result of companies not utilising Agile coaches, or utilising very junior or unskilled coaches.

Higher utilisation of Agile coaches implies that a company that wants to embark on an Agile transformation can, from the onset, avoid 36% of the challenges if they were to employ a suitably skilled Agile coach(es).

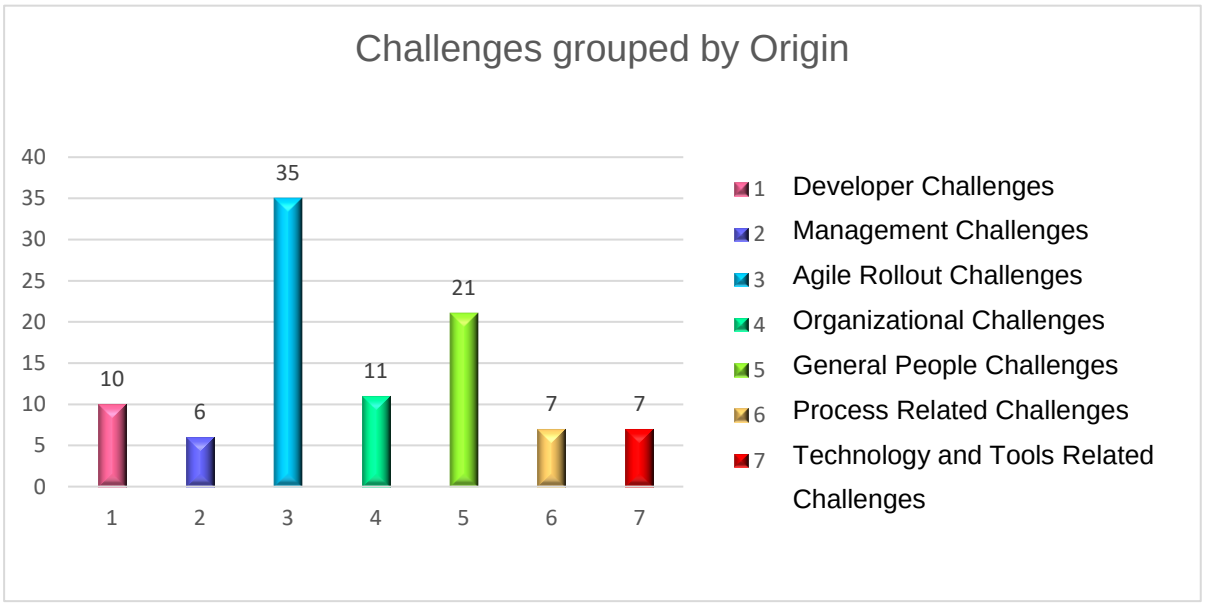


Figure 2.2: Normalised challenges grouped by Origin

2.4.3.2 Categories / Types of challenges

Besides the originating areas for the challenges discussed above, specific general categories can further be distinguished for the challenges (Dikert, Paasivaara & Lassenius, 2016). These categories are listed in Table 2.22 below.

As can be seen in Figure 2.3 below, “Agile is difficult to implement” is the most significant category of challenges by a considerable margin. Companies ought to understand the difficulty of undergoing an Agile transformation, and should subsequently not embark on it without a proper assessment to see if they are prepared for it, and prepared to invest in it (Nerur, Mahapatra & Mangalaraj, 2005;77). A vital part of investment ought to be towards senior Agile coaches, as indicated above, including addressing Agile roll-out challenges.

Category 5 (Different approaches emerge in a multi-team environment) did not get any challenges allocated towards it after normalisation was applied to the challenges. The challenges assigned to it was in one instance deemed as redundant in Table 2.23, awarded toward “Agile difficult to implement” in the next or “Coordination challenges in the multi-team environment” in another.

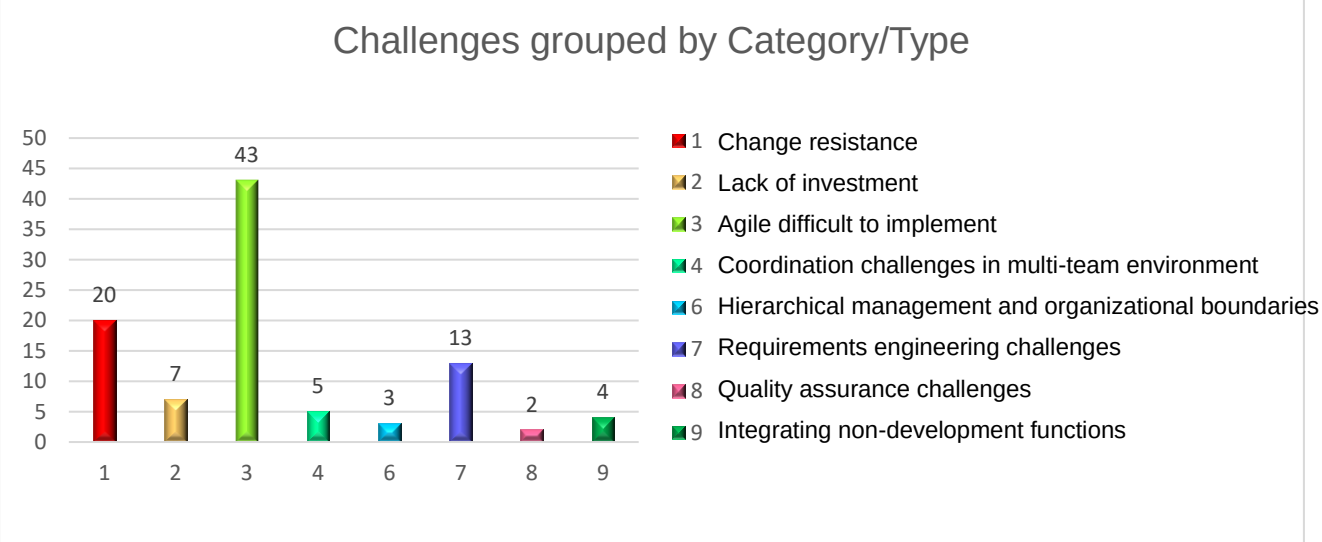


Figure 2.3: Challenges grouped by category

ID	Challenge Categories
1	Change resistance

ID	Challenge Categories
2	Lack of investment
3	Agile difficult to implement
4	Coordination challenges in the multi-team environment
5	Different approaches emerge in a multi-team environment
6	Hierarchical management and organisational boundaries
7	Requirements engineering challenges
8	Quality assurance challenges
9	Integrating non-development functions

Table 2.22: The types or categories of challenges, as mentioned by Dikert, Paasivaara and Lassenius (2016)**.**

2.4.3.3 Normalised challenges

To ensure that the normalised Agile adoption challenges listed below are understandable, the following "statement check" for each of the Agile adoption challenges has been created:

"We are struggling to progress with Agile adoption because (of a/the) [*fill in challenge*]"

If the "statement check" is not understandable, it is removed from the list or reworded to highlight out the *true* meaning of the challenge.

For the normalisation process to be successful, each challenge listed by the various authors referenced had to be very well understood to ensure the correct assignment and categorisation. In most instances, normalised challenges were renamed if there were two or more duplicates discovered so that it speaks to the previous challenges.

The following items were excluded from the challenges list, and the various motivations for doing so are included:

Reasons challenge not added	Challenges	Reference

It is a result, not a challenge in itself. E.g., This could be due to a lack of Agile coaching.	Reverting to the old way of working	(Dikert, Paasivaara & Lassenius, 2016)
It is a result, not a challenge in itself. E.g., This could be due to a lack of Agile coaching.	Interpretation of agile differs between teams	(Dikert, Paasivaara & Lassenius, 2016)
It is a result, not a challenge in itself. E.g., This could be due to a lack of Agile coaching.	Inconsistent agile practices and process	(Versionone, 2016)
Specific to Scrum	Scrum Implementation (They found the mindset shift quite hard to adapt to.)	(Hajjdiab & Taleb, 2011a:8-9)
Specific to Scrum	Scrum Implementation (They found the mindset shift quite hard to adapt to.)	(Hajjdiab & Taleb, 2011b)
It is a result not a challenge in itself. E.g. This could be caused by a lack of Agile coaching, lack of guidelines or collaboration.	Diverged agile models between teams	(Pitkänen, 2015)
Not a challenge as Agile methods is better than most methods for getting rid of waiting for design. Agile focuses on simple emerging design. (Beck et al., 2001)	Complex projects: Waiting times in the process, specifically in design waiting for requirements	(Mishra & Mishra, 2011:551)
Not a challenge as Agile methods is better than most traditional methods of getting to higher test coverage through practices like TDD and pair programming. (Agile Alliance, 2001)	Reduction of test coverage due to a shortage of projects and lack of independent testing	(Mishra & Mishra, 2011:551)
This is not an Agile adoption challenge. It is a general challenge for the team to get the Scrum Master to focus on process as well. It is not a factor not to adopt Agile. If the SM issue persists, the SM will need to be replaced.	The Scrum Master is caring only about the individuals and interactions (and ignoring the process).	(Marchenko & Abrahamsson, 2008)
This is an issue in any method and not only for Agile so it should not stop Agile Adoption	Non-development functions: Identification (non-functional requirements)	(Almeida, 2017)
Specific to Scrum	Placing an overemphasis on the Scrum process and practices.	(Marchenko & Abrahamsson, 2008)

Table 2.23: Challenges excluded from the final normalised list.

Besides the challenges and references column, three other meaningful columns are present:

- Solution Exist? → Indicates whether through the literature review a clear indication is provided for a possible solution to the challenge.
- Category → "Category" is discussed prior and can be referenced in Table 2.22.
- Severity → If an author indicated a severity value for a specific challenge, it was added; otherwise, it was left out. Only 21.6% of the challenges in the selected literature have a severity rating provided by the relevant authors.

None of the authors provided a difficulty rating for any of their challenges, and it was therefore not added to the table below. Without both the difficulty and severity of each challenge, it would be impossible to determine a priority for a specific challenge.

86.6% of challenges have at least one possible solution mapped to it, which left thirteen challenges unmapped. The list of mapped solutions was completed by interrogating the various success factors in the following section.

ID	Challenges	Solution Exist?	Category	Severity	References
	Developer Challenges				
1	Developers fear their skill deficiencies will be more transparent.	TRUE	1		(Conboy et al., 2010:50)
2	There is a perceived need for developers to be a "master of all trades."	TRUE	1		(Conboy et al., 2010:50)
3	There is an increased reliance on social skills: e.g. Pair Programming and deeper need for collaboration.	TRUE	1		(Conboy et al., 2010:50), (Gandomani et al., 2013), (Noruwana & Tanner, 2012)
4	Lack of business knowledge among developers	TRUE	1		(Conboy et al., 2010:50)
5	Practices are prioritised over the Agile values and principles, and we do not understand it well enough.	TRUE	1		(Conboy et al., 2010:50), (Dikert, Paasivaara & Lassenius, 2016), (Pitkänen, 2015)
6	Lack of developer motivation to use agile methods (Developer resistance)	TRUE	1		(Conboy et al., 2010:50)
7	The implications of devolved decision-making (Team makes decisions)	TRUE	1		(Conboy et al., 2010:50)
8	Lack of agile-specific recruitment policies and suitably trained IT graduates	TRUE	2		(Conboy et al., 2010:50)
9	The perceived rewards for Using Agile Techniques are minimal.	FALSE	3	2.05	(Vijayasarathy & Turk, 2008)

10	Lack of big design up front.	FALSE	3	3.00	(Rodríguez et al., 2012), (Mishra & Mishra, 2011:551)
	Management Challenges				
11	Lack of management support	TRUE	1	3.51	(Vijayasarathy & Turk, 2008), (Versionone, 2016), (Hajjdiab & Taleb, 2011a:8-9), (Rodríguez et al., 2012), (Hajjdiab & Taleb, 2011b), (Scrumology, 2009), (Marchenko & Abrahamsson, 2008)
12	Management is unwilling to change (waterfall mode; Command and Control)	FALSE	1	2.80	(Dikert, Paasivaara & Lassenius, 2016), (Gandomani et al., 2013), (Nerur, Mahapatra & Mangalaraj, 2005), (Rodríguez et al., 2012)
13	It can be difficult to convince management of the need for a new development approach.	FALSE	3		(Scrumology, 2009)
14	Management wants to combine elements of RUP and Agile.	FALSE	3		(Scrumology, 2009)
15	Management is interfering too much.	FALSE	1		(Marchenko & Abrahamsson, 2008)
16	Individual and team resistance to change. The main reason is the fear of losing a job (Gandomani et al., 2014)	FALSE	1		(Almeida, 2017)
	Agile Rollout Challenges				
17	Lack of training	FALSE	2	3.18	(Vijayasarathy & Turk, 2008), (Dikert, Paasivaara & Lassenius, 2016), (Versionone, 2016), (Srinivasan & Lundqvist, 2009), (Rodríguez et al., 2012), (Almeida, 2017)
18	Lack of formal guidelines on Agile especially Quality, Scaling and practices	TRUE	3	2.46	(Vijayasarathy & Turk, 2008), (Dikert, Paasivaara & Lassenius, 2016), (Noruwana & Tanner, 2012), (Rodríguez et al., 2012), (Pitkänen, 2015), (Almeida, 2017)
19	There is a lack of change management during the adoption process	TRUE	3		(Dikert, Paasivaara & Lassenius, 2016), (Boehm & Turner, 2005)
20	Lack of coaching and mentorship.	FALSE	2		(Dikert, Paasivaara & Lassenius, 2016), (Hajjdiab & Taleb, 2011a:8-9), (Srinivasan & Lundqvist, 2009), (Hajjdiab & Taleb, 2011b)
21	There is a perception of too high a workload. Balancing an Agile project with other traditional projects.	FALSE	3		(Dikert, Paasivaara & Lassenius, 2016), (Hajjdiab & Taleb, 2011b), (Hajjdiab & Taleb, 2011a:8-9)

22	Previous Agile attempts were customised poorly.	FALSE	3		(Dikert, Paasivaara & Lassenius, 2016)
23	Certain teams and individuals are overzealous	FALSE	3		(Dikert, Paasivaara & Lassenius, 2016), (Hajjdiab & Taleb, 2011a:8-9)
24	Coordination and management of work across teams are hard.	TRUE	4		(Dikert, Paasivaara & Lassenius, 2016), (Pitkänen, 2015), (Mishra & Mishra, 2011:551), (Almeida, 2017)
25	The perception that high-level requirements management is mostly missing in Agile.	FALSE	7		(Dikert, Paasivaara & Lassenius, 2016)
26	Requirements refinement are challenging.	FALSE	7		(Dikert, Paasivaara & Lassenius, 2016)
27	The creation and estimation of user stories are hard.	FALSE	3		(Dikert, Paasivaara & Lassenius, 2016), (Almeida, 2017)
28	The perceived gap between long and short-term planning (Fitzgerald & Stol, 2014)	FALSE	7		(Dikert, Paasivaara & Lassenius, 2016)
29	Non-functional testing (performance security, scalability, etc.) is hard for large projects.	TRUE	8		(Dikert, Paasivaara & Lassenius, 2016), (Pitkänen, 2015)
30	Lack of automated testing	FALSE	8		(Dikert, Paasivaara & Lassenius, 2016)
31	The perception that requirements are conflicting over multiple teams and sprints. This has a severe impact on QA	FALSE	7		(Dikert, Paasivaara & Lassenius, 2016), (Almeida, 2017)
32	Short, iterative and incremental deliveries are a challenge.	TRUE	3	3.50	(Dikert, Paasivaara & Lassenius, 2016), (Nerur, Mahapatra & Mangalaraj, 2005), (Rodríguez et al., 2012), (Marchenko & Abrahamsson, 2008)
33	Product launch activities cannot be easily adjusted.	FALSE	9		(Dikert, Paasivaara & Lassenius, 2016)
34	Traditional need to overgenerate documentation. The perception that it is a safety net.	TRUE	3	3.20	(Gandomani et al., 2013), (Nerur, Mahapatra & Mangalaraj, 2005), (Hajjdiab & Taleb, 2011a:8-9), (Rodríguez et al., 2012), (Hajjdiab & Taleb, 2011b), (Almeida, 2017)
35	Absence of a Pilot Project	TRUE	3		(Hajjdiab & Taleb, 2011a:8-9), (Hajjdiab & Taleb, 2011b)
36	Requirements are difficult to identify and to have proper traceability to end product.	FALSE	7		(Srinivasan & Lundqvist, 2009), (Almeida, 2017)
37	Measuring Agile success is difficult.	FALSE	1	3.60	(Rodríguez et al., 2012)
38	Defining business value is difficult.	FALSE	7	3.90	(Rodríguez et al., 2012)
39	Tailoring the appropriate Agile/Lean practices are difficult	FALSE	3	3.50	(Rodríguez et al., 2012)
40	The perception that predictability has decreased in Agile.	FALSE	3	3.10	(Rodríguez et al., 2012), (Schwaber, Laganza & D'Silva, 2007)

41	Lack of understanding of how to include user-centred design people and processes in large-scale Agile projects.	TRUE	3		(Pitkänen, 2015) (Sintes, 2017)
42	Requirements management with large/complex products is difficult.	TRUE	7		(Pitkänen, 2015), (Mishra & Mishra, 2011:551)
43	Continuous testing surfaces a lot of practical issues.	FALSE	3		(Mishra & Mishra, 2011:551)
44	The perception that maintenance effort has increased with an increase in the number of releases; undermining team productivity and morale.	TRUE	3		(Mishra & Mishra, 2011:551), (Marchenko & Abrahamsson, 2008)
45	Requirements priority lists are hard to create and maintain.	FALSE	7		(Mishra & Mishra, 2011:551)
46	Variance on the speed of delivery between Agile and traditional running concurrently and having interdependencies.)	TRUE	3		(Boehm & Turner, 2005)
47	Legacy systems do not integrate with Agile methods easily.	TRUE	7		(Boehm & Turner, 2005)
48	Teams are split, and critical resources are removed after a successful pilot.	TRUE	3		(Boehm & Turner, 2005)
49	Team members are overly specialised, which requires work to be handed off several times.	FALSE	3		(Scrumology, 2009)
50	Agile, quality concepts are not in place and not well understood.	TRUE	3		(Scrumology, 2009) (Almeida, 2017)
51	Teams are committing to too much.	FALSE	3		(Marchenko & Abrahamsson, 2008)
	Organisational Challenges				
52	There is a general organisational resistance to change	TRUE	1	3.73	(Vijayasarathy & Turk, 2008), (Dikert, Paasivaara & Lassenius, 2016), (Noruwana & Tanner, 2012), (VersionOne, 2016), (Rodríguez et al., 2012), (Pitkänen, 2015), (Almeida, 2017)
53	Co-location challenges are hard to resolve.	TRUE	2		(Dikert, Paasivaara & Lassenius, 2016), (Boehm & Turner, 2005), (Almeida, 2017)
54	The autonomous team model is challenging (self-organisation).	FALSE	4		(Dikert, Paasivaara & Lassenius, 2016)
55	Communication over distributed teams is challenging.	TRUE	4		(Dikert, Paasivaara & Lassenius, 2016), (Gandomani et al., 2013)

56	Company philosophy or culture at odds with core agile values. Bureaucratic with too much governance and compliance.	TRUE	1	3.80	(Dikert, Paasivaara & Lassenius, 2016), (Nerur, Mahapatra & Mangalaraj, 2005), (Noruwana & Tanner, 2012), (VersionOne, 2016), (Hajjdiab & Taleb, 2011a:8-9), (Rodríguez et al., 2012), (Hajjdiab & Taleb, 2011b), (Almeida, 2017)
57	Internal silos are kept. (Lindstrom & Jeffries, 2004)	FALSE	6		(Dikert, Paasivaara & Lassenius, 2016), (Almeida, 2017)
58	Other functions in the company are unwilling to change	TRUE	9		(Dikert, Paasivaara & Lassenius, 2016), (Pitkänen, 2015)
59	The complexity of moving from a traditional to an Agile approach.	TRUE	3		(Gandomani et al., 2013), (Nerur, Mahapatra & Mangalaraj, 2005), (VersionOne, 2016), (Boehm & Turner, 2005)
60	Contracts are fixed priced.	FALSE	3	3.30	(Rodríguez et al., 2012)
61	Process standard ratings like ISO and CMMI levels are impacted.	TRUE	3		(Boehm & Turner, 2005)
62	Legislation; the rules, procedures and external audits are stifling the flow.	FALSE	6		(Almeida, 2017)
	General People Challenges				
63	There is a need for agile-compliant performance evaluation	TRUE	3		(Conboy et al., 2010:50), (Dikert, Paasivaara & Lassenius, 2016), (Nerur, Mahapatra & Mangalaraj, 2005), (Noruwana & Tanner, 2012), (Boehm & Turner, 2005), (Scrumology, 2009)
64	Lack of Peer Support	FALSE	1	2.96	(Vijayasarathy & Turk, 2008)
65	There is scepticism towards the new way of working	FALSE	1		(Dikert, Paasivaara & Lassenius, 2016)
66	Middle managers do not know what role they play in Agile.	TRUE	6		(Dikert, Paasivaara & Lassenius, 2016), (Gandomani et al., 2013), (Boehm & Turner, 2005), (Almeida, 2017)
67	Cultural differences in multi-international teams.	FALSE	4		(Gandomani et al., 2013)
68	Customer or client requirement to be part of the team.	TRUE	9		(Gandomani et al., 2013), (Nerur, Mahapatra & Mangalaraj, 2005)
69	Team members, in general, find it difficult to work effectively in a team.	FALSE	3		(Nerur, Mahapatra & Mangalaraj, 2005), (Scrumology, 2009), (Almeida, 2017)
70	The perception that Agile demands a high level of competence.	FALSE	1		(Nerur, Mahapatra & Mangalaraj, 2005)
71	Specialised skills are needed; like refactoring, configuration management, developer unit testing (TDD), and various others.	FALSE	1	3.10	(Nerur, Mahapatra & Mangalaraj, 2005), (Rodríguez et al., 2012), (Mishra & Mishra, 2011:551), (Almeida, 2017)

72	Team members find it difficult to change the roles they are used to.	FALSE	1		(Noruwana & Tanner, 2012)
73	Lack of previous experience with agile methods	FALSE	3		(VersionOne, 2016), (Almeida, 2017)
74	The role of a single identifiable Product Owner is missing.	FALSE	3		(VersionOne, 2016), (Srinivasan & Lundqvist, 2009), (Scrumology, 2009)
75	Collaboration in and between teams as well as with stakeholders are ineffective	TRUE	3	3.65	(VersionOne, 2016), (Rodríguez et al., 2012), (Marchenko & Abrahamsson, 2008), (Almeida, 2017)
76	Steep learning curve.	FALSE	3	3.10	(Rodríguez et al., 2012)
77	Lack of ownership by the team.	FALSE	3		(Scrumology, 2009)
78	The Scrum Master refuses to protect the team.	TRUE	1		(Scrumology, 2009)
79	Possible frictions between teams.	TRUE	4		(Scrumology, 2009)
80	The focus is more on the individuals than the team (Over-individualism).	TRUE	3		(Marchenko & Abrahamsson, 2008)
81	Availability and time consumption of training courses are worrisome.	FALSE	2		(Almeida, 2017)
82	The identification of real customer needs is difficult.	FALSE	7		(Almeida, 2017)
83	The customer can introduce changes as much as he or she wants.	FALSE	7		(Almeida, 2017)
	Process Related Challenges				
84	Integration and technical consistency are very hard to achieve over multi-interdependent areas.	FALSE	7		(Dikert, Paasivaara & Lassenius, 2016), (Almeida, 2017)
85	Choosing an appropriate Agile method is hard.	FALSE	3		(Gandomani et al., 2013), (Nerur, Mahapatra & Mangalaraj, 2005)
86	The uncertainty that Agile can scale appropriately for large projects.	FALSE	3	3.60	(Nerur, Mahapatra & Mangalaraj, 2005), (Rodríguez et al., 2012)
87	Inherent emphasis on non-functional requirements.	FALSE	9		(Almeida, 2017)
88	Quality assurance requirement of multiple reviews and inspections.	FALSE	3		(Almeida, 2017)
89	Agile Risk management concerning identification, prioritisation and management is not explicit.	FALSE	3		(Almeida, 2017)
90	Tracking velocity in a scaled environment is not clear.	FALSE	3		(Almeida, 2017) (Cohn, 2005)
	Technology and Tools Related Challenges				
91	Using non-flexible tools and hardware is a barrier to moving to agile.	FALSE	2		(Gandomani et al., 2013)

92	Existing tools are inappropriate.	TRUE	2	3.10	(Gandomani et al., 2013), (Nerur, Mahapatra & Mangalaraj, 2005), (VersionOne, 2016), (Rodríguez et al., 2012), (Scrumology, 2009)
93	Interdependency management between various systems is hard and adds to the complexity of the architecture.	TRUE	7		(Boehm & Turner, 2005)
94	Techniques and reporting in Agile to measure progress is difficult.	TRUE	3		(Boehm & Turner, 2005), (Marchenko & Abrahamsson, 2008), (Almeida, 2017)
95	There is not a proper continuous integration or build system and process in place.	TRUE	3		(Scrumology, 2009) (Almeida, 2017)
96	Coordination complexities between multiple product owners in large-scale projects.	FALSE	3		(Almeida, 2017)
97	Issue tracking and traceability in Agile is difficult.	FALSE	3		(Almeida, 2017)

Table 2.24: Normalised Challenges from selected literature

2.4.4 **Conclusion**

Answers to the following research sub-questions are provided in section 2.4:

a. What are the general unique lists of Agile adoption challenges and success factors in literature? See section 2.4.3.3 and 2.5.
b. What is the relationship between the various Agile adoption challenges originating areas? See section 2.4.3.1.
c. Are all industries doing software development similarly impacted by Agile adoption challenges? → In this study no clear answer is ascertained from the material referenced. It is only Dikert, Paasivaara and Lassenius' (2016) study where a mapping between industries (Table 2.25) is present. Their results for the challenges listed are distributed and varied among a few industries. Some literature (VersionOne, 2016) indicates industries, but not as a mapping to individual challenges. The conclusion in this study is that either a specific industry does not have a particular mapping to specific challenges (with Agile adoption challenges which are for the most part similar across industries), or there is not sufficient research to conclude whether there is a definitive mapping between industries and challenges.

d. What are the various other variables describing Agile adoption challenges in Agile software development? The answer to this question will assist in building a comprehensive filter for the automated Agile adoption coach. → The only other distinguishing factor about this question can be found in the various categories discussed in section 2.4.3.2.

ID	Software industry	Financial Services	Prof. Serv.	Health care	Govern ment	Telecom	Retail	Media Entertain.	Internet Services	Origin	Cate gory
13	True	False	True	False	False	True	False	False	False	4	1
14	True	True	True	False	False	True	False	False	False	2	1
15	False	False	False	False	False	True	False	True	True	3	2
16	False	False	True	False	False	True	True	False	False	5	1
17	True	True	True	False	False	True	False	False	False	3	3
18	True	False	True	False	True	False	False	False	False	1	3
19	False	True	True	False	False	True	False	False	True	5	1
20	True	False	True	False	False	True	False	False	True	3	3
21	False	True	True	False	False	False	False	False	False	2	1
22	False	False	True	False	True	True	False	True	True	3	2
23	True	False	True	False	False	True	False	False	False	3	3
24	True	False	True	False	False	True	False	False	False	4	2
25	True	True	True	False	False	True	False	False	True	3	3
26	False	False	True	True	False	True	False	False	True	3	3
27	True	True	False	False	False	True	True	False	True	3	4
28	False	False	False	False	True	True	False	False	False	4	4
29	False	False	True	False	False	False	False	False	False	4	4
30	True	False	True	False	False	False	False	False	True	6	7
31	True	False	True	False	False	True	False	False	True	5	6
32	True	False	True	False	False	False	False	False	False	4	1
33	True	False	False	False	False	True	False	False	False	4	6
34	True	True	True	False	False	True	False	False	True	3	7
35	True	True	True	False	False	True	False	False	False	3	7
36	True	False	False	False	False	False	False	False	True	3	3
37	False	False	False	True	False	True	True	False	True	3	7
38	True	False	True	True	True	True	False	False	False	3	8

Table 2.25: Mapping between industries and challenges by Diken, Paasivaara & Lassenius (2016)

2.5 Success Factors and Solutions

2.5.1 Introduction

Several studies focus on the antithesis of Agile adoption challenges.

1. Misra, Kumar & Kumar, 2009	2. Dikert, Paasivaara & Lassenius, 2016
3. Chow & Cao, 2008	4. Gorans & Kruchten, 2016
5. Pitkänen, 2015	6. Kropp, Martin & Meier, 2015
7. Bavani, 2009	8. Nguyen, 2016
9. Brown, 2013	10. Boehm & Turner, 2005

Table 2.26: Studies mentioning specific success factors

The texts in Table 2.26 highlight the **success factors** in making teams and projects successful. Most of these factors can further be translated into possible solutions for Agile adoption challenges if the mappings to the challenges are clearly identified in the literature. The solutions to these challenges are, however, not explicitly stated, as can already be seen in section 2.4 that speaks to adoption challenges. This study is based on an accepted mapping between challenges and solutions, which led to an automated framework to determine possible solutions to challenges quickly. From the "Challenges" section above we have found six main contributors to possible Agile adoption solutions (specifically towards a mapping between challenges and solutions): (Boehm & Turner, 2005; Scrumology, 2009; Conboy et al., 2010; Hajjdiab & Taleb, 2011b; Gandomani et al., 2013; Dikert, Paasivaara & Lassenius, 2016). These six studies contribute to 110 solutions that have mappings to challenges. The other 24 solutions listed in Table 6.2 are described as solutions, but no clear challenges are mentioned. Table 6.2 contains all 262 unnormalised success factors and solutions found in the literature. Table 2.27 list all the literature referenced to gain clearly stated solutions to Agile adoption challenges.

(Almeida, 2017)	(Kakar, 2017)
(Anderson, 2003)	(Lindstrom & Jeffries, 2004)
(Beck et al., 2001)	(Marchenko & Abrahamsson, 2008)
(Boehm & Turner, 2005)	(Nguyen, 2016)
(Chow & Cao, 2008)	(Nivoit, 2013)
(Cohn, 2005)	(O'Connor & Duchonova, 2014)
(Conboy et al., 2010)	(Ramesh, Cao & Baskerville, 2010)
(De Haaff, 2017)	(Schwaber, Laganza & D'Silva, 2007)
(Dikert, Paasivaara & Lassenius, 2016)	(Scrumology, 2009)
(Fitzgerald & Stol, 2014)	(Sidky, 2007)
(Gandomani et al., 2013)	(Sintes, 2017)
(Hajjdiab & Taleb, 2011b)	

Table 2.27: Literature indicating Agile adoption solutions

Most Agile coaches would incorporate a set of Agile success factors in an Agile adoption/transformation framework to apply for a given client - either by utilising a framework in Table 2.1 or incorporating a custom framework (ORiordan, 2017 and Prokhorenko, 2012:191). Most of these frameworks are initiated by some assessment to determine the real scope and challenges of the client. It is therefore not sufficient merely to know success factors. One instead has to understand what success factors or solutions can be applied to specific challenges.

In this section, we will review the various success factors of the ten studies mentioned in Table 2.26 above, and together with the solutions offered in the literature, devise a normalised list of success factors and solutions that will assist in mapping between challenges and solutions. Table 2.28 presents a normalised list of success factors and solutions to the studies presented in Table 3.5.

In this study success factors in respect to Agile adoption and Agile method execution are defined as unique factors of success, and do not necessarily incorporate every individual practice and principle of Agile or any of the Agile methods as a success factor.

2.5.2 Existing Success factors' studies

Brown (2013:121) is interested in the success factors that speak to Agile user experience designers, and therefore, his success factor focus is narrower. The success factors mentioned are very generalised, speaking broadly to project over the process focus, team dynamics, communication, envisioning the big picture and training. Under each of these high-level success factors, Brown elaborates tremendously on the topics with various examples.

Nguyen (2016:175-179) identifies five general success factors and various sub-factors which affect the success of Agile software development teams. He incorporates a lot from the work of Chow and Cao (2008) concerning the main success factor groups, except "Technologies and Development Tools". Under "process factors" he lists most of the Scrum processes under various phases. These "success factors" in themselves, according to the definition in 6.4.1 of success factors, cannot be listed as success factors

in this dissertation per se. The factors listed by Chow and Cao (2008), however, which he referenced, can be utilised.

Misra, Kumar and Kumar (2009:1874) follow a hybrid approach of quantitative and qualitative analysis, with most of the information gathered by quantitative analysis. This analysis is quite significant as it covers multiple industries and organisation sizes from practitioners across the world. The authors divide the success factors into the two main categories of people-related and organisational factors. The authors further subdivide them into 13 other success factors.

Chow and Cao (2008:969-970) determine through a literature review that there are five dimensions of success factors speaking to 36 related success factors. After extensive quantitative research on Agile projects, the authors devise a new categorisation of the success factors. They find that delivery strategy, Agile software engineering techniques, team capability, project management process, team environment and customer involvement are the critical success factor categories. They further establish that delivery strategy, Agile software engineering techniques and team capability are the most important. Under this new set of categories, 25 success factors are indicated. Of these 25, only one additional factor is added, namely “Good progress tracking mechanism”.

The 29 success factors mentioned by Dikert, Paasivaara and Lassenius (2016) are organised into 11 categories. They do not follow the typical division of success factors found in other literature, such as that of (Nguyen, 2016), (Misra, Kumar & Kumar, 2009) and (Chow & Cao, 2008) which divide success factors into people, organisational, technical, process and project factors.

In the comparative literature review of Pitkänen (2015), ten success factors are identified as prominent and which can be found in at least three other studies. The success factors are not categorised or prioritised in any way but do add value to the overall list of possible success factors.

Paul Gorans speaks to his 31 years of experience when he lists his selection of 10 Agile success factors (Gorans & Kruchten, 2016:3). After each success factor mentioned, the

authors succinctly explain in bullet points the motivation behind the factor's importance. They do not, however, offer any categorisation.

The ten success factors in Bavani (2009) are experienced-based and mostly focus on Agile success factors for distributed Agile teams. No categories are given for the factors. A drawback of this study is that some of the examples are not always in-line with Agile best practices. In the explanation of the factor "*Take Stock of User Stories for Status Checks"*, for example, it ought logically and explicitly to have involved a product owner, however in the study it only refers to coordination between project managers and Agile leaders. This study provides a new, highly helpful and specific overview of success factors for distributed teams, which is not as straightforward as co-located teams.

Kropp, Martin and Meier (2015) present a qualitative study involving eight successful Agile organisations. They devise nine success factors which they organise under the categories of engineering practices, management and organisation practices for success and Agile values. They further notice successful teams always adapt to meet complexity in challenges, and their success factors consequently also adapt.

As mentioned under the section "*Challenges*", Boehm and Turner (2005) provide 21 general suggestions to mitigate challenges under the categories of the development process, business process and people. These suggestions are not necessarily success factors, but they are instrumental in addressing challenges under the categories above. Their study is furthermore focused on environments where there is a transition from a traditional to an Agile state, and where there can still be an in-between hybrid state where two approaches need to co-exist.

2.5.3 **Discussion**

The unnormalised list of success factors and solutions total 263 in various categories. After normalisation is applied to the list (removing duplicates and ambiguities) and following the four-step process as laid out in Table 3.13, the list totals 143 possible solutions. The author expected significantly more convergence to a shorter list of possible solutions; however, there were ultimately merely a few factors that were very common over all the selected studies. Training, communication, pilot project, Agile method

customisation, Agile performance reviews and customer commitment are the most common factors with six and five authors mentioning it respectively, as can be seen in Table 2.28. Four authors cite factors 7 to 11 in Table 2.28 and 12 to 28 by three authors. Eighty-one factors are unique among the studies, despite five of these studies (Chow & Cao, 2008), (Misra, Kumar & Kumar, 2009), (Pitkänen, 2015), (Dikert, Paasivaara & Lassenius, 2016) and (Nguyen, 2016) completing significant literature reviews on the subject matter. This sparse list of success factors could be the result of the above authors not attempting to create an all-inclusive list, but merely referencing those factors that were of the most important to them.

It is important to note that the normalised list of success factors and possible solutions mostly indicate an idea or a reference to a possible solution. The inclination is to point the reader in the generally correct direction for further research after they have followed the mapping from the Agile challenge of interest. A one-liner solution statement would rarely be enough to elaborate satisfactorily on a specific, complex challenge.

These success factors or possible solutions to Agile adoption challenges below are listed without their respective categories (which is supplied by the authors mentioned in Table 3.5). The factors' categories will assist with the mapping; however, it is not essential as after the mapping between challenges and solutions are done, the assigned categories of the challenges is utilised.

#	Solutions	Reference
1	Educate on the need of training investment and provide Agile and Method training for all teams and stakeholders, including new roles.	(Pitkänen, 2015; Misra, Kumar & Kumar, 2009; Nguyen, 2016; Boehm & Turner, 2005; Dikert, Paasivaara & Lassenius, 2016; Brown, 2013)
2	A pilot project must be selected when first adopting agile, and it should be near the middle of what is the average for an organisation, small enough to be done by one team and should not be critical to the organisation	(Hajjdiab & Taleb, 2011b; Boehm & Turner, 2005; Dikert, Paasivaara & Lassenius, 2016; Chow & Cao, 2008; Pitkänen, 2015)
3	One of the valuable benefits of agile development is its ability to be customised based on the culture and the environment of the organisation it is adopted in. These changes should not compromise its beneficiary and affect its efficiency. It might not be realistic to change the whole organisation's system overnight, but that does not mean that systems should not be revised, and enhanced with what is best for the organisation.	(Hajjdiab & Taleb, 2011b; Chow & Cao, 2008; Nguyen, 2016; Misra, Kumar & Kumar, 2009; Kropp, Martin & Meier, 2015)
4	Long-term solutions are dependent on rewarding teamwork and breadth of understanding. While there is no single or easy solution, some ideas include having team goals and bonuses, eliminating	(Scrumology, 2009; Gorans & Kruchten, 2016; Boehm & Turner,

	individual performance reviews, and allowing teams the opportunity to make mistakes.	2005; Chow & Cao, 2008; Nguyen, 2016)
5	More communication and negotiation between people. Emphasis on face-to-face.	(Brown, 2013; Nguyen, 2016; Misra, Kumar & Kumar, 2009; Chow & Cao, 2008; Kropp, Martin & Meier, 2015)
6	Good customer relationship and commitment (involvement) leading to greater customer satisfaction.	(Kropp, Martin & Meier, 2015; Chow & Cao, 2008; Nguyen, 2016; Misra, Kumar & Kumar, 2009; Dikert, Paasivaara & Lassenius, 2016)
7	Ensure multiple members get agile training or attend agile conferences.	(Conboy et al., 2010; Brown, 2013; Pitkänen, 2015; Misra, Kumar & Kumar, 2009)
8	Setup more closely co-located project teams and explaining the value of it.	(Nguyen, 2016; Misra, Kumar & Kumar, 2009; Chow & Cao, 2008; Nivoit, 2013)
9	Explain the value of coherent, self-organising teamwork.	(Nguyen, 2016; Dikert, Paasivaara & Lassenius, 2016; Kakar, 2017; Chow & Cao, 2008)
10	Following agile-oriented requirement management process.	(Chow & Cao, 2008; Dikert, Paasivaara & Lassenius, 2016; Kropp, Martin & Meier, 2015; Nguyen, 2016)
11	Select team members with high competence and expertise.	(Chow & Cao, 2008; Misra, Kumar & Kumar, 2009; Gorans & Kruchten, 2016; Nguyen, 2016)
12	Mature Agile coaching and championing.	(Conboy et al., 2010; Kropp, Martin & Meier, 2015; Gorans & Kruchten, 2016)
13	Build a sharing and learning environment to empower team decision-making.	(Conboy et al., 2010; Bavani, 2009; Misra, Kumar & Kumar, 2009)
14	Teach managers to an Agile style of leadership and collaboration.	(Gandomani et al., 2013; Chow & Cao, 2008; Nguyen, 2016)
15	In agile software engineering literature, the mature Agile coach has a critical role to play to ensure successful outcomes. An organisation needs to hire an Agile coach for at least six months to increases the chance of succeeding in the Agile adoption process.	(Hajjdiab & Taleb, 2011b; Dikert, Paasivaara & Lassenius, 2016; Pitkänen, 2015)
16	In the situation where a single customer cannot be identified, pinpoint the project sponsor, who is the single person who ultimately approves the funding of the project. By working with the sponsor and clearly articulating the need for a single business representative on the project team, this situation can usually be quickly resolved.	(Scrumology, 2009; Nguyen, 2016; Chow & Cao, 2008)
17	Motivate, educate and train teams in the use of Test-driven development (TDD), Pair programming, continuous integration (CI) and refactoring as starters.	(Scrumology, 2009; Chow & Cao, 2008; Nguyen, 2016)
18	Provide appropriate technical training to the team.	(Chow & Cao, 2008; Nguyen, 2016; Dikert, Paasivaara & Lassenius, 2016)
19	Ensure management support.	(Dikert, Paasivaara & Lassenius, 2016; Chow & Cao, 2008; Nguyen, 2016)
20	Customise the agile approach carefully (small gradual improvements).	(Dikert, Paasivaara & Lassenius, 2016; Pitkänen, 2015; Boehm & Turner, 2005)
21	Cut Communication Loops in distributed teams (timely resolution).	(Bavani, 2009; Misra, Kumar & Kumar, 2009; Nguyen, 2016)

22	Recognise the importance of the Product Owner role and getting it right.	(Dikert, Paasivaara & Lassenius, 2016; Pitkänen, 2015; Gorans & Kruchten, 2016)
23	If multiple releases are followed, both long and short-term planning is done continuously. Otherwise, long-term planning is done in the initiation/inception/discovery phase.	(Fitzgerald & Stol, 2014; Brown, 2013; Dikert, Paasivaara & Lassenius, 2016)
24	Planning for specific technologies and tools are essential for collaboration, reporting and effectiveness.	(Gorans & Kruchten, 2016; Nguyen, 2016; Bavani, 2009)
25	More critical is personal characteristics: Team members with attributes like collaborative attitude, honesty, sense of responsibility, eagerness to learn, and willingness to work with others are just as prominent as high skill level.	(Brown, 2013; Misra, Kumar & Kumar, 2009; Nguyen, 2016)
26	Apply throughput accounting rather than cost accounting in software development projects. (Cost accounting comes from manufacturing whereas throughput accounting focuses on delivered value.) (Anderson, 2003)	(Anderson, 2003; Boehm & Turner, 2005; Chow & Cao, 2008)
27	Educate on need of pursuing simple design.	(Dikert, Paasivaara & Lassenius, 2016; Nguyen, 2016; Misra, Kumar & Kumar, 2009)
28	Conducting “Just Enough” (good preparation) upfront work before the start of the Agile project (Design included - detail design in development).	(Gorans & Kruchten, 2016; Bavani, 2009; Boehm & Turner, 2005)
29	Customer company runs training sessions on essential Agile topics within the business domain and the company-specific area(s).	(Conboy et al., 2010; Dikert, Paasivaara & Lassenius, 2016)
30	Ensure cross-team observation/validation of agile practices.	(Conboy et al., 2010; Pitkänen, 2015)
31	Documentation should be just enough. Define appropriate knowledge management strategy and distribution of knowledge in different levels of the organisation.	(Gandomani et al., 2013; Chow & Cao, 2008)
32	Upper management approval is fundamental to support any major change in the processes of any department; without their approval, the success of the adoption is doubted. To acquire the upper management support a presentation or a meeting might be conducted to demonstrate the new agile development and explain the benefits of changing to a new method.	(Hajjdiab & Taleb, 2011b; Dikert, Paasivaara & Lassenius, 2016)
33	Let the teamwork directly with the customer	(Scrumology, 2009; Misra, Kumar & Kumar, 2009)
34	Management requires education, coaxing and convincing to recognise that there is a better way to develop software.	(Scrumology, 2009; Dikert, Paasivaara & Lassenius, 2016)
35	When working with management, pay attention to both Agile metrics (or the lack, thereof), and adaptive planning over predictive planning. The Agile approach to both of these is counter-intuitive for most classically trained managers and requires constant reinforcing.	(Scrumology, 2009; Brown, 2013)
36	Educate management and business partners on a single approach and the value of it.	(Dikert, Paasivaara & Lassenius, 2016; Scrumology, 2009)
37	Implementing Test-Driven Development (TDD) and Continuous Integration (CI) are reasonable first steps. The tools for both of these practices are standard and (often) free.	(Scrumology, 2009; Kropp, Martin & Meier, 2015)
38	Utilising a maturity and risk assessment of the organisation will help to determine the main resistance criteria and how to scale. Tools like the AAC, 4 stage process (identifying discontinuing factors) and SAMI of Sidky (2007) might be useful.	(Sidky, 2007; Chow & Cao, 2008)
39	Managers who have a light-touch or adaptive management style.	(Chow & Cao, 2008; Nguyen, 2016)

40	Educate and train on Automated testing & code-level unit testing.	(Scrumology, 2009; Kropp, Martin & Meier, 2015)
41	Communicate the change intensively.	(Dikert, Paasivaara & Lassenius, 2016; Pitkänen, 2015)
42	Make the change transparent and open.	(Dikert, Paasivaara & Lassenius, 2016; Kropp, Martin & Meier, 2015)
43	Create and communicate positive experiences and celebrate small wins.	(Dikert, Paasivaara & Lassenius, 2016; Bavani, 2009)
44	Implement change leadership and recognise the importance of it (Leadership owning the change initiative).	(Pitkänen, 2015; Dikert, Paasivaara & Lassenius, 2016)
45	Educate on the value of regular delivery of software. Speaking of core Agile principles.	(Chow & Cao, 2008; Nguyen, 2016)
46	Cultural training and facilitation aiming at cultural differences in media utilisation and communication could also evidence beneficial for Agile software development functioning.	(Nguyen, 2016; Chow & Cao, 2008)
47	Good progress tracking mechanism that communicates well.	(Chow & Cao, 2008; Gorans & Kruchten, 2016)
48	Create a competency model for team members to assess personal development paths.	(Misra, Kumar & Kumar, 2009; Nguyen, 2016)
49	Planning and Control - Agile way. Informal internalised plans and qualitative control.	(Misra, Kumar & Kumar, 2009; Nguyen, 2016)
50	Cherish agile communities of practices.	(Dikert, Paasivaara & Lassenius, 2016; Pitkänen, 2015)
51	Investigate and update contracting practices to support agile concepts.	(Boehm & Turner, 2005; Gorans & Kruchten, 2016)
52	Waterfall gives a perception of predictability, but as it is mostly unempirical, it is less predictable. Empirical, incremental and iterative Agile models utilising velocity provides more predictability.	(Schwaber, Laganza & D'Silva, 2007; Boehm & Turner, 2005)
53	Testing - executable test cases determine the success of requirements and testing.	(Kropp, Martin & Meier, 2015; Nguyen, 2016)
54	Emphasise value. Value of team, and members of the team staying together and prioritised requirements.	(Boehm & Turner, 2005; Chow & Cao, 2008)
55	Explain that Agile values to people and that it is part of the core values. Agile transformation is about transforming into a new mindset that can benefit the organisation and individuals.	(Beck et al., 2001; Dikert, Paasivaara & Lassenius, 2016)
56	Allow the customer to have full authority.	(Nguyen, 2016; Misra, Kumar & Kumar, 2009)
57	Utilising the correct tools traceability in Agile among defects, user stories, test cases, and how it lands up in production, should not be difficult. It is essential the whole team understands how to map the work items and the need for it.	(Almeida, 2017; Bavani, 2009)
58	More quality control procedures agreed upon and administered by the project team. It is needed as quality should be a given in Agile.	(Bavani, 2009; Chow & Cao, 2008)
59	Educate on the smaller size of project teams.	(Misra, Kumar & Kumar, 2009; Chow & Cao, 2008)
60	Support to attain a facility with the proper agile-style work environment.	(Chow & Cao, 2008; Nguyen, 2016)
61	Following agile-oriented configuration management process	(Chow & Cao, 2008; Nguyen, 2016)

62	Maintenance on multiple releases may increase, but two things should be kept in mind: Agile in itself demands high-quality deliveries. Otherwise, multiple releases will not be sustainable and secondly satisfying the customer through early, and continuous releases are core in Agile and should be a key motivator for the team.	(Beck et al., 2001; Nivoit, 2013)
63	Feedback outside stand-ups, allowing the documentation of any fears, issues or concerns inappropriate for discussion in open forum	(Conboy et al., 2010)
64	Stand up meetings voluntary for new junior developers.	(Conboy et al., 2010)
65	Dedicated mentor for new staff.	(Conboy et al., 2010)
66	Weaker developers paired with those who had more experience, taking joint responsibility for requirements.	(Conboy et al., 2010)
67	Use pair programming and pair rotation to distribute knowledge and facilitate learning.	(Conboy et al., 2010)
68	Encourage task self-assignment to allow the developer to work in different areas and learn new skills.	(Conboy et al., 2010)
69	Reintroduce specific roles when it is perceived beneficial to teams with, e.g. large team size, conflicts between developers	(Conboy et al., 2010)
70	Combine development and training program to provide customised training materials on social skills, using developers' examples.	(Conboy et al., 2010)
71	Using proper documentation to back up communication.	(Conboy et al., 2010)
72	Provide small training modules (on a frequent basis), making it interactive to allow developers to acquire niche business knowledge required by the project	(Conboy et al., 2010)
73	Recruit staff and graduates with a combination of IT and business knowledge	(Conboy et al., 2010)
74	Assess agility regarding agile values and principles, not practice adherence.	(Conboy et al., 2010)
75	Try to have multiple 'bought-in' developers on each team.	(Conboy et al., 2010)
76	Collecting and sharing successful adoption stories and positive experiences.	(Conboy et al., 2010)
77	Implement a democratic voting system.	(Conboy et al., 2010)
78	Project manager plays the role of facilitator.	(Conboy et al., 2010)
79	Performance evaluation needs to consider the breadth of skills, not just depth.	(Conboy et al., 2010)
80	Performance evaluation to apply much higher weighting for mentoring, voluntary contributions etc.	(Conboy et al., 2010)
81	360° feedback is necessary.	(Conboy et al., 2010)
82	Develop specific recruiting practices tailored to agile methods to hire the right people.	(Conboy et al., 2010)
83	Use team recruiting to find the right person working in the team.	(Conboy et al., 2010)
84	Put newly recruited graduates on agile projects to get hands-on experience.	(Conboy et al., 2010)
85	Guide managers to follow a director and coordinator model.	(Gandomani et al., 2013)
86	Changing mindset takes time and managers should be continually mentored and coached.	(Gandomani et al., 2013)
87	Coaches should patiently coach the team on the value of the customer in the team.	(Gandomani et al., 2013)

88	Select appropriate personnel and provide the necessary training, mentoring and creating a set of work practices that promote process excellence.	(Gandomani et al., 2013)
89	Companies should use tools that can supply incremental evolution, continuous integration, re-working, version management and support other agile practices.	(Gandomani et al., 2013)
90	Invest in tools that support distributed agile teams.	(Gandomani et al., 2013)
91	Careful planning and discipline of agile adoption are required. A team must also expect a slight decrease in productivity when first adopting agile while the teams learn the new implementation techniques.	(Hajjdiab & Taleb, 2011b)
92	The adoption of agile should be planned in a time that has the minimum work pressure, for instance, a company should invest more time by not accepting new projects for six months and investing this time only for agile adoption.	(Hajjdiab & Taleb, 2011b)
93	Many organisations complain about the lack of documentation, but also over-documenting and spending plenty of time documenting unnecessary information is incorrect. What agile development offers is a new way of comprehensive documentation that takes less time and effort. This is another issue that needs to be supported by the upper management and agreed upon with the customers from the beginning of the project.	(Hajjdiab & Taleb, 2011b)
94	To minimise hand-offs, encourage pairing at all stages and ensure that the necessary handoffs are informal. Encouraging the team to tackle small chunks of work also helps because it forces them to communicate more frequently.	(Scrumology, 2009)
95	Encourage team members to own and solve their problems instead of letting organisations control technical conversations.	(Scrumology, 2009)
96	Solicit input from different members of the team.	(Scrumology, 2009)
97	Every member of the team needs to contribute, and if one or two members of the team are putting themselves above the team, this needs to be quickly addressed by management. Often, this results in the quick removal of the individual(s).	(Scrumology, 2009)
98	The first step in educating an organisation about Agile development is speaking to the team or individual members of the team. However, you also need to consider a wider audience, including functional managers; the PMO and HR. Failure to address this wider audience can hobble the transition to Agile.	(Scrumology, 2009)
99	Agile teams must frequently communicate with senior management to clearly articulate their successes and challenges, and they should share their experiences with the broader organisation, so their achievements are acknowledged.	(Scrumology, 2009)
100	In the situation where several different groups have an equal interest in the success of a project, the team still needs a single representative who is willing to work with each of the different groups and prioritise accordingly. Again, work with the project sponsor to provide the path to a solution.	(Scrumology, 2009)
101	If the Scrum Master is ineffective at protecting the team, this person needs to be replaced. To find a new Scrum Master, request volunteers or ask the team whom they would like to represent them.	(Scrumology, 2009)
102	The Scrum Master and project team need to address the issue with senior management. They need to explain why specific tools are relevant and need to be supported, and they need to make management understand that having the right tools can make the	(Scrumology, 2009)

	difference between a good enough product and a high-quality product.	
103	The standard approach is to prioritise dependant functionality early and to code to an agreed-upon the interface, but this is overly simplistic. Instead, teams need to recognise that there is a dynamic relationship between dependant teams that need to be actively and continuously improved.	(Scrumology, 2009)
104	Evaluating risks through an Agile assessment is the best overall approach to determining how much agility is enough in a transforming environment.	(Boehm & Turner, 2005)
105	Establish guidelines for safe and agility-compatible process maturity assessments where ISO or CMMI cannot be followed.	(Boehm & Turner, 2005)
106	Engage everyone in the organisation.	(Dikert, Paasivaara & Lassenius, 2016)
107	Identify incompatible assumptions (model clashes) and synergies between agile and traditional methods within your organisational processes.	(Boehm & Turner, 2005)
108	Understand how communication occurs within development teams.	(Boehm & Turner, 2005)
109	Make management support visible.	(Dikert, Paasivaara & Lassenius, 2016)
110	Communicate that change is non-negotiable.	(Dikert, Paasivaara & Lassenius, 2016)
111	Define specific functionality or responsibilities that you are going to address with agile approaches. (Specific Agile focus or large-scale adoption?)	(Boehm & Turner, 2005)
112	Educate the organisation on the value of Agile coaching and the risks of not doing it.	(O'Connor & Duchonova, 2014)
113	Ensure Explicit Delegation (no misunderstanding) and Validate Assumptions for distributed teams.	(Bavani, 2009)
114	Initiate Test Drives (all teams do testing on the integrated product).	(Bavani, 2009)
115	Allow grass roots level empowerment.	(Dikert, Paasivaara & Lassenius, 2016)
116	Try setup projects with no multiple independent teams.	(Chow & Cao, 2008)
117	Focus on the XP core practice of whole team focus and educate on the value of it.	(Lindstrom & Jeffries, 2004)
118	High-level requirements upfront in Agile are not missing. It just lacks the in-depth detail in the initial state. Explain that detailed requirements will be developed during the development phase.	(Ramesh, Cao & Baskerville, 2010)
119	Translate agile and software issues into management and customer language. Describe in terms that the audience can connect with.	(Boehm & Turner, 2005)
120	While delivering in short release cycles ensure that non-software areas (e.g. marketing) are up to date on progress, in order not to cause bottlenecks and release disappointments. All stakeholders on project need to be Agile, not only software development.	(Lindstrom & Jeffries, 2004)
121	Conduct empirical studies of which change are more unpredictable and therefore suited for agile and which are more predictable and suitable for traditional plan-driven methods.	(Boehm & Turner, 2005)
122	Using existing knowledge and not reinventing the wheel.	(Gorans & Kruchten, 2016)

123	Firstly, you need to understand the function of the various mainstream Agile methods currently. You can also evaluate by looking at Project size, Team size, Iteration length, Roles and responsibilities, Virtual team support, Risk mitigation level, Customer interaction and Pros and cons.	(De Haaff, 2017)
124	Include persons with previous agile experience.	(Dikert, Paasivaara & Lassenius, 2016)
125	Setup a structured approach to facilitate inter-team communication.	(Pitkänen, 2015)
126	Nurturing craftsmanship. (The idea of becoming the best you can be.)	(Kropp, Martin & Meier, 2015)
127	In Agile, "Just in Time" and "Sprint Pairs" are the most common UX models. Just in Time - (1. Design done within a sprint. 2. Typically requires predefined and commonly used chunks of design. 3. Requires a lot of collaboration. 4. Iterative with parallel efforts and predicted throwaway work. 5. Easier to track.) and Sprint Pairs - (1. Design works a sprint ahead of development. 2. The scope is traded off for designing within the time box. 3. Respects functional dependencies.)	(Sintes, 2017)
128	Develop architectures that support the compartmentalisation of agile and traditional teams, where the hybrid approach is in place.	(Boehm & Turner, 2005)
129	Develop management and architectural practices for hybrid agile and plan-driven methods.	(Boehm & Turner, 2005)
130	While the whole team is not fully Agile, and there exist specific interdependencies between various method teams, key upfront agreements and risk acceptance have to be understood and agreed upon. This approach requires much discipline and takes away from the productivities a full Agile team could have achieved.	(Boehm & Turner, 2005)
131	Research on how to modify or reconceive legacy systems to enable and help agility-compatible re-engineering and maintenance, replacement, or extension.	(Boehm & Turner, 2005)
132	It is imperative that the whole team need to work well together. Every team requires a mature Scrum Master or Agile leader that will help to facilitate that. All team members' work should be evident.	(Marchenko & Abrahamsson, 2008)
133	Committing to only what is entirely sure and instead pull in more work when all work is done early.	(Marchenko & Abrahamsson, 2008)
134	Educate management on self-organisation and that the customer needs are vital to the team's success. They will not ignore it. Management will have to trust them.	(Marchenko & Abrahamsson, 2008)
135	Investigate to see what is the absolute mandatory legislative and audit requirements. Agree with management on only making provision for that in the customised method.	(Almeida, 2017)
136	In large-scale projects or programs, product owners need to collaborate a lot, and the uses of a scrum of scrums for product owners as well as an overarching Product owner can help with this challenge.	(Almeida, 2017)
137	Agile is asking for end-to-end working software at the end of every increment. All non-functional requirements are inclusive of that. Significant rework can be introduced later on if it is not addressed early enough.	(Almeida, 2017)
138	Manage Effort Variance Constructively (issues causing a variance between different teams).	(Bavani, 2009)

139	At the start of the project, it is good if various teams can agree on an estimation method and sizes for reference stories. E.g., Everyone understands what a user story size of three looks like. They all use it from there on as a reference. This way, all teams will have their velocity but based on the same referencing system.	(Cohn, 2005)
140	When testing needs (functional or non-functional) overlap between different teams, a separate specialised QA team can be established. These people need to be well trained in their area. Coordination between QA and development teams need to be defined.	(Dikert, Paasivaara & Lassenius, 2016)
141	Arrange social events	(Dikert, Paasivaara & Lassenius, 2016)
142	Integrating critical speciality skills to support Agile teams	(Gorans & Kruchten, 2016)
143	Pragmatic customisation of the agile process	(Pitkänen, 2015)

Table 2.28: Normalised success factors and solutions

2.6 Mapping Solutions to Challenges

The mapping of challenges to possible solutions is provided in Table 2.29. This mapping answers question 1f: "What is the mapping between Agile adoption challenges and recorded possible solutions to these challenges in the Agile software development space?"

The mapping of normalised solutions (Table 2.28) to challenges (Table 2.24) in Table 2.29 below is merely a modest attempt at reflecting the possibilities mentioned in the literature, as well as from the author's personal experience. The literature contains single (one to one) mappings to 84 of the 97 challenges - 86.6% of the challenges are covered by at least one solution per challenge. The literature does not contain direct mappings to 13 of the 97 challenges. Additional solutions are mapped subjectively to the challenges until no more solutions are left. The AAC will be a living database that will be kept up to date for the benefit of the Agile community.

Challenge	Solutions
The absence of a Pilot Project	A pilot project must be selected when first adopting agile, and it should be near the middle of what is the average for an organisation, small enough to be done by one team and should not be critical to the organisation.
Agile quality concepts are not in place and not well understood.	Motivate, educate and train teams in the use of Test-driven development (TDD), Pair programming, continuous integration (CI) and refactoring as starters.
Agile Risk management concerning identification, prioritisation and management is not explicit.	Utilising a maturity and risk assessment of the organisation will help to determine the main resistance criteria and how to scale. Tools like the AAC, 4 stage process (identifying discontinuing factors) and SAMI of Sidky (2007) might be useful.
Agile Risk management concerning identification, prioritisation and management is not explicit.	In agile software engineering literature, the mature Agile coach has a critical role to play to ensure successful outcomes. An organisation needs to hire an Agile coach for at least six months to increases the chance of succeeding in the Agile adoption process.

Availability and time consumption of training courses are worrisome.	Educate on the need of training investment and provide Agile and Method training for all teams and stakeholders, including new roles.
Certain teams and individuals are overzealous	Careful planning and discipline of agile adoption are required. A team must also expect a slight decrease in productivity when first adopting agile while the teams learn the new implementation techniques.
	Mature Agile coaching and championing.
	Combine development and training program to provide customised training materials on social skills, using developers' examples.
	Select appropriate personnel and provide the necessary training, mentoring and creating a set of work practices that promote process excellence.
	Educate on the need of training investment and provide Agile and Method training for all teams and stakeholders, including new roles.
Choosing an appropriate Agile method is hard.	Using existing knowledge and not reinventing the wheel.
	Identify incompatible assumptions (model clashes) and synergies between agile and traditional methods within your organisational processes.
	Educate management and business partners on a single approach and the value of it.
	Firstly, you need to understand the function of the various mainstream Agile methods currently. You can also evaluate by looking at Project size, Team size, Iteration length, Roles and responsibilities, Virtual team support, Risk mitigation level, Customer interaction and Pros and cons.
Collaboration in and between teams as well as with stakeholders are ineffective	Setup a structured approach to facilitate inter-team communication.
	Ensure Explicit Delegation (no misunderstanding) and Validate Assumptions for distributed teams.
	Invest in tools that support distributed agile teams.
	Good progress tracking mechanism that communicates well.
	Cut Communication Loops in distributed teams (timely resolution).
	Understand how communication occurs within development teams.
	Agile teams must frequently communicate with senior management to clearly articulate their successes and challenges, and they should share their experiences with the broader organisation, so their achievements are acknowledged.
	Planning for specific technologies and tools are essential for collaboration, reporting and effectiveness.
	Teach managers to an Agile style of leadership and collaboration.
	More communication and negotiation between people. Emphasis on face-to-face.
Co-location challenges are hard to resolve.	Setup more closely co-located project teams and explaining the value of it.
	Educate on the smaller size of project teams.
	Understand how communication occurs within development teams.
	Support to attain a facility with the proper agile-style work environment.
Communication over distributed teams is challenging.	Understand how communication occurs within development teams.
	Cut Communication Loops in distributed teams (timely resolution).
	Cultural training and facilitation aiming at cultural differences in media utilisation and communication could also evidence beneficial for Agile software development functioning.
	Setup a structured approach to facilitate inter-team communication.
	More communication and negotiation between people. Emphasis on face-to-face.
	Ensure Explicit Delegation (no misunderstanding) and Validate Assumptions for distributed teams.
	Invest in tools that support distributed agile teams.
	Agile teams must frequently communicate with senior management to clearly articulate their successes and challenges, and they should share their experiences with the broader organisation, so their achievements are acknowledged.
	Try setup projects with no multiple independent teams.

	At the start of the project, it is good if various teams can agree on an estimation method and sizes for reference stories. E.g., Everyone understands what a user story size of three looks like. They all use it from there on as a reference. This way, all teams will have their velocity based on the same referencing system.
Company philosophy or culture at odds with core agile values. Bureaucratic with too much governance and compliance.	One of the valuable benefits of agile development is its ability to be customised based on the culture and the environment of the organisation it is adopted in. These changes should not compromise its beneficiary and affect its efficiency. It might not be realistic to change the whole organisation system overnight, but that does not mean that systems should not be revised and enhanced to what is best for the organisation.
	Utilising a maturity and risk assessment of the organisation will help to determine the main resistance criteria and how to scale. Tools like the AAC, 4 stage process (identifying discontinuing factors) and SAMI of Sidky (2007) might be useful.
	Assess agility regarding agile values and principles, not practice adherence.
	Customise the agile approach carefully (small gradual improvements).
	One of the valuable benefits of agile development is its ability to be customised based on the culture and the environment of the organisation it is adopted in. These changes should not compromise its beneficiary and affect its efficiency. It might not be realistic to change the whole organisation system overnight, but that does not mean that systems should not be revised and enhanced to what is best for the organisation.
The complexity of moving from a traditional to an Agile approach.	Conduct empirical studies of which change are more unpredictable and therefore suited for agile and which are more predictable and suitable for traditional plan-driven methods.
	Translate agile and software issues into management and customer language. Describe in terms that the audience can connect with.
	Establish guidelines for safe and agility-compatible process maturity assessments where ISO or CMMI cannot be followed.
	Identify incompatible assumptions (model clashes) and synergies between agile and traditional methods within your organisational processes.
	Evaluating risks through an Agile assessment is the best overall approach to determining how much agility is enough in a transforming environment.
Continuous testing surfaces a lot of practical issues.	Testing - executable test cases determine the success of requirements and testing.
	Implementing Test-Driven Development (TDD) and Continuous Integration (CI) are reasonable first steps. The tools for both of these practices are standard and (often) free.
	Motivate, educate and train teams in the use of Test-driven development (TDD), Pair programming, continuous integration (CI) and refactoring as starters.
	Initiate Test Drives (all teams do testing on the integrated product).
	Educate and train on Automated testing & code-level unit testing.
Contracts are fixed priced.	Investigate and update contracting practices to support agile concepts.
Coordination complexities between multiple product owners in large-scale projects.	In large-scale projects or programs, product owners need to collaborate a lot, and the uses of a scrum of scrums for product owners as well as an overarching Product owner can help with this challenge.
Coordination and management of work across teams are hard.	Ensure Explicit Delegation (no misunderstanding) and Validate Assumptions for distributed teams.
	Invest in tools that support distributed agile teams.
	Cut Communication Loops in distributed teams (timely resolution).
	Initiate Test Drives (all teams do testing on the integrated product).
	Understand how communication occurs within development teams.
Cultural differences in multi-international teams.	Cultural training and facilitation aiming at cultural differences in media utilisation and communication could also evidence beneficial for Agile software development functioning.
	Good progress tracking mechanism that communicates well.
	Invest in tools that support distributed agile teams.

	Planning for specific technologies and tools are essential for collaboration, reporting and effectiveness.
Customer or client requirement to be part of the team.	Coaches should patiently coach the team on the value of the customer in the team.
Defining business value is difficult.	Apply throughput accounting rather than cost accounting in software development projects. (Cost accounting comes from manufacturing whereas throughput accounting focuses on delivered value.) (Anderson, 2003)
	Educate on the value of regular delivery of software. Speaking to core Agile principles.
	Recognise the importance of the Product Owner role and getting it right.
	Following agile-oriented requirement management process.
Developers fear their skill deficiencies will be more transparent.	Feedback outside stand-ups, allowing the documentation of any fears, issues or concerns inappropriate for discussion in open forum.
	Stand up meetings voluntary for new junior developers.
	Dedicated mentor for new staff.
	Weaker developers paired with those who had more experience, taking joint responsibility for requirements.
Existing tools are inappropriate.	Invest in tools that support distributed agile teams.
	The Scrum Master and project team need to address the issue with senior management. They need to explain why specific tools are relevant and need to be supported, and they need to make management understand that having the right tools can make the difference between a good enough product and a high-quality product.
Individual and team resistance to change, the main reason is the fear of losing a job (Gandomani et al., 2014)	Explain that Agile values to people and that it is part of the core values. Agile transformation is about transforming into a new mindset that can benefit the organisation and individuals.
Inherent emphasis on non-functional requirements.	Agile is asking for end-to-end working software at the end of every increment. All non-functional requirements are inclusive of that. Significant rework can be introduced later on if it is not addressed early enough.
Integration and technical consistency are very hard to achieve over multi-interdependent areas.	Try setup projects with no multiple independent teams.
	Companies should use tools that can supply incremental evolution, continuous integration, re-working, version management and support other agile practices.
	Motivate, educate and train teams in the use of Test-driven development (TDD), Pair programming, continuous integration (CI) and refactoring as starters.
	Initiate Test Drives (all teams do testing on the integrated product).
	Invest in tools that support distributed agile teams.
	Ensure Explicit Delegation (no misunderstanding) and Validate Assumptions for distributed teams.
Interdependency management between various systems is hard and adds to the complexity of the architecture.	The standard approach is to prioritise dependant functionality early and to code to an agreed-upon the interface, but this is overly simplistic. Instead, teams need to recognise that there is a dynamic relationship between dependant teams that need to be actively and continuously improved.
	Try setup projects with no multiple independent teams.
Internal silos are kept.	Focus on the XP core practice of whole team focus and educate on the value of it.
	One of the valuable benefits of agile development is its ability to be customised based on the culture and the environment of the organisation it is adopted in. These changes should not compromise its beneficiary and affect its efficiency. It might not be realistic to change the whole organisation system overnight, but that does not mean that systems should not be revised and enhanced to what is best for the organisation.
	Make management support visible.
	Communicate that change is non-negotiable.
	Assess agility regarding agile values and principles, not practice adherence.
Issue tracking and traceability in Agile are difficult.	Utilising the correct tools traceability in Agile among defects, user stories, test cases, and how it lands up in production, should not be difficult. It is essential the whole team understands how to map the work items and the need for it.

It can be difficult to convince management of the need for a new development approach.	The first step in educating an organisation about Agile development is speaking to the team or individual members of the team. However, you also need to consider a wider audience, including functional managers; the PMO and HR. Failure to address this wider audience can hobble the transition to Agile.
	Management requires education, coaxing and convincing to recognise that there is a better way to develop software.
	When working with management, pay attention to both Agile metrics (or the lack, thereof), and adaptive planning over predictive planning. The Agile approach to both of these is counter-intuitive for most classically trained managers and requires constant reinforcing.
Lack of agile-specific recruitment policies and suitably trained IT graduates	Develop specific recruiting practices tailored to agile methods to hire the right people.
	Use team recruiting to find the right person working in the team.
	Put newly recruited graduates on agile projects to get hands-on experience.
Lack of automated testing	Educate and train on Automated testing & code-level unit testing.
	Implementing Test-Driven Development (TDD) and Continuous Integration (CI) are reasonable first steps. The tools for both of these practices are standard and (often) free.
	Motivate, educate and train teams in the use of Test-driven development (TDD), Pair programming, continuous integration (CI) and refactoring as starters.
	Educate on the need of pursuing simple design.
	Conducting “Just Enough” (proper preparation) upfront work before the start of the Agile project (Design included - detail design in development).
Lack of business knowledge among developers	Customer company runs training sessions on essential Agile topics within the business domain and the company-specific area(s).
	Provide small training modules (on a frequent basis), making it interactive to allow developers to acquire niche business knowledge required by the project
	Recruit staff and graduates with a combination of IT and business knowledge
Lack of coaching and mentorship.	In agile software engineering literature, the mature Agile coach has a critical role to play to ensure successful outcomes. An organisation needs to hire an Agile coach for at least six months to increases the chance of succeeding in the Agile adoption process.
	Educate the organisation on the value of Agile coaching and the risks of not doing it.
	Dedicated mentor for new staff.
	Performance evaluation to apply much higher weighting for mentoring and voluntary contributions.
	Changing mindset takes time and managers should be continually mentored and coached.
	Select appropriate personnel and provide the necessary training, mentoring and creating a set of work practices that promote process excellence.
	Mature Agile coaching and championing.
Lack of developer motivation to use agile methods (Developer resistance)	Arrange social events
	Try to have multiple ‘bought-in’ developers on each team.
	Collecting and sharing successful adoption stories and positive experiences.
Lack of formal guidelines on Agile especially Quality, Scaling and practices	Mature Agile coaching and championing.
	In agile software engineering literature, the mature Agile coach has a critical role to play to ensure successful outcomes. An organisation needs to hire an Agile coach for at least six months to increases the chance of succeeding in the Agile adoption process.
	Define specific functionality or responsibilities that you are going to address with agile approaches. (Specific Agile focus or large-scale adoption?)
Lack of management support	Upper management approval is fundamental to support any significant change in the processes of any department; without their approval, the success of the adoption is doubted. To acquire the upper management support a presentation or a meeting might be conducted to demonstrate the new agile development and explain the benefits of changing to a new method.

	Agile teams must frequently communicate with senior management to clearly articulate their successes and challenges, and they should share their experiences with the broader organisation, so their achievements are acknowledged.
Lack of ownership by the team.	Let the teamwork directly with the customer
	Encourage team members to own and solve their problems instead of letting organisations control technical conversations.
	Solicit input from different members of the team.
Lack of Peer Support	Understand how communication occurs within development teams.
	Ensure management support.
	Make management support visible.
	Communicate that change is non-negotiable.
	Evaluating risks through an Agile assessment is the best overall approach to determining how much agility is enough in a transforming environment.
	Try to have multiple 'bought-in' developers on each team.
Lack of previous experience with agile methods	Include persons with previous agile experience.
	Educate on the need of training investment and provide Agile and Method training for all teams and stakeholders, including new roles.
	Provide appropriate technical training to the team.
	Ensure multiple members get agile training or attend agile conferences.
	Mature Agile coaching and championing.
	Educate the organisation on the value of Agile coaching and the risks of not doing it.
	Changing mindset takes time and managers should be continually mentored and coached.
	In agile software engineering literature, the mature Agile coach has a critical role to play to ensure successful outcomes. An organisation needs to hire an Agile coach for at least six months to increases the chance of succeeding in the Agile adoption process.
	Create and communicate positive experiences and celebrate small wins.
Lack of training	Educate on the need of training investment and provide Agile and Method training for all teams and stakeholders, including new roles.
	Provide appropriate technical training to the team.
Lack of understanding of how to include user-centred design people and processes in large-scale Agile projects.	In Agile, "Just in Time" and "Sprint Pairs" are the most common UX models. Just in Time - (1. Design done within a sprint. 2. Typically requires predefined and commonly used chunks of design. 3. Requires a lot of collaboration. 4. Iterative with parallel efforts and predicted throwaway work. 5. Easier to track.) And Sprint Pairs - (1. Design works a sprint ahead of development. 2. The scope is traded off for designing within the time box. 3. Respects functional dependencies.)
Legacy systems do not integrate with Agile methods easily.	Research on how to modify or reconceive legacy systems to enable and help agility-compatible re-engineering and maintenance, replacement, or extension.
Legislation; the rules, procedures and external audits are stifling the flow.	Investigate to see what the absolute mandatory legislative and audit requirements are. Agree with management on only making provision for that in the customised method.
Management is interfering too much.	Educate management on self-organisation and that the customer needs are vital to the team's success. They will not ignore it. Management will have to trust them.
	Explain the value of coherent, self-organising teamwork.
Management is unwilling to change (waterfall mode; Command and Control)	Teach managers to an Agile style of leadership and collaboration.
	Changing mindset takes time and managers should be continually mentored and coached.
Management wants to combine elements of RUP and Agile.	Educate management and business partners on a single approach and the value of it.
Measuring Agile success is difficult.	Planning and Control - Agile way. Informal internalised plans and qualitative control.
	When working with management, pay attention to both Agile metrics (or the lack, thereof), and adaptive planning over predictive planning. The Agile approach to

	both of these is counter-intuitive for most classically trained managers and requires constant reinforcing.
	Good progress tracking mechanism that communicates well.
	Planning for specific technologies and tools are essential for collaboration, reporting and effectiveness.
	Companies should use tools that can supply incremental evolution, continuous integration, re-working, version management and support other agile practices.
	Invest in tools that support distributed agile teams.
	Apply throughput accounting rather than cost accounting in software development projects. (Cost accounting comes from manufacturing whereas throughput accounting focuses on delivered value.) (Anderson, 2003)
Middle managers do not know what role they play in Agile.	Guide managers to follow a director and coordinator model.
	Project manager plays the role of facilitator.
	Teach managers to an Agile style of leadership and collaboration.
	Guide managers to follow a director and coordinator model.
	Changing mindset takes time and managers should be continually mentored and coached.
	Managers who have a light-touch or adaptive management style.
Non-functional testing (performance security, scalability, etc.) is hard for large projects.	When testing needs (functional or non-functional) overlap between different teams, a separate specialised QA team can be established. These people need to be well trained in their area. Coordination between QA and development teams need to be defined.
	Agile is asking for end-to-end working software at the end of every increment. All non-functional requirements are inclusive of that. Significant rework can be introduced later on if it is not addressed early enough.
	Educate on the need of training investment and provide Agile and Method training for all teams and stakeholders, including new roles.
Other functions in the company are unwilling to change	Communicate the change intensively.
	Make the change transparent and open.
	Create and communicate positive experiences and celebrate small wins.
	Implement change leadership and recognise the importance of it (Leadership owning the change initiative).
	Translate agile and software issues into management and customer language. Describe in terms that the audience can connect with.
	Communicate that change is non-negotiable.
	Ensure management support.
	Make management support visible.
Perception of too high workload. Balancing an Agile project with other traditional projects.	The adoption of agile should be planned in a time that has the minimum work pressure, for instance, a company should invest more time by not accepting new projects for six months and investing this time only for agile adoption.
The perception that Agile demands a high level of competence.	Select team members with high competence and expertise.
	Create a competency model for team members to assess personal development paths.
	More critical is personal characteristics: Team members with attributes like collaborative attitude, honesty, sense of responsibility, eagerness to learn, and willingness to work with others are just as prominent as high skill level.
	Nurturing craftsmanship. (The idea of becoming the best you can be.)
The perception that high-level requirements management is mostly missing in Agile.	Following agile-oriented requirement management process.
	High-level requirements upfront in Agile are not missing. It just lacks the in-depth detail in the initial state. Explain that detailed requirements will be developed during the development phase.
	Mature Agile coaching and championing.

The perception that maintenance effort has increased with an increase in the number of releases; undermining team productivity and morale.	Maintenance on multiple releases may increase, but two things should be kept in mind: Agile in itself demands high-quality deliveries. Otherwise, multiple releases will not be sustainable and secondly satisfying the customer through early, and continuous releases are core in Agile and should be a key motivator for the team.
The perception that predictability has decreased in Agile.	Waterfall gives a perception of predictability, but as it is mostly unempirical, it is less predictable. Empirical, incremental and iterative Agile models utilising velocity provides more predictability.
	When working with management, pay attention to both Agile metrics (or the lack, thereof), and adaptive planning over predictive planning. The Agile approach to both of these is counter-intuitive for most classically trained managers and requires constant reinforcing.
	In agile software engineering literature, the mature Agile coach has a critical role to play to ensure successful outcomes. An organisation needs to hire an Agile coach for at least six months to increases the chance of succeeding in the Agile adoption process.
The perception that requirements are conflicting over multiple teams and sprints. This has a severe impact on QA	The standard approach is to prioritise dependant functionality early and to code to an agreed-upon the interface, but this is overly simplistic. Instead, teams need to recognise that there is a dynamic relationship between dependant teams that need to be actively and continuously improved.
	Invest in tools that support distributed agile teams.
	Recognise the importance of the Product Owner role and getting it right.
Possible frictions between teams.	The standard approach is to prioritise dependant functionality early and to code to an agreed-upon the interface, but this is overly simplistic. Instead, teams need to recognise that there is a dynamic relationship between dependant teams that need to be actively and continuously improved.
Practices are prioritised over the Agile values and principles, and we do not understand it well enough.	Ensure multiple members get agile training or attend agile conferences.
	Mature Agile coaching and championing.
	Ensure cross-team observation/validation of agile practices.
Practices are prioritised over the Agile values and principles, and we do not understand it well enough.	Assess agility regarding agile values and principles, not practice adherence.
Previous Agile attempts were customised poorly.	In agile software engineering literature, the mature Agile coach has a critical role to play to ensure successful outcomes. An organisation needs to hire an Agile coach for at least six months to increases the chance of succeeding in the Agile adoption process.
	Mature Agile coaching and championing.
	Utilising a maturity and risk assessment of the organisation will help to determine the main resistance criteria and how to scale. Tools like the AAC, 4 stage process (identifying discontinuing factors) and SAMI of Sidky (2007) might be useful.
	Evaluating risks through an Agile assessment is the best overall approach to determining how much agility is enough in a transforming environment.
	Establish guidelines for safe and agility-compatible process maturity assessments where ISO or CMMI cannot be followed.
	Customise the agile approach carefully (small gradual improvements).
Process standard ratings like ISO and CMMI levels are impacted.	Establish guidelines for safe and agility-compatible process maturity assessments where ISO or CMMI cannot be followed.
Product launch activities cannot be easily adjusted.	While delivering in short release cycles ensure that non-software areas (e.g. marketing) are up to date on progress, in order not to cause bottlenecks and release disappointments. All stakeholders on project need to be Agile, not only software development.
Quality assurance requirement of multiple reviews and inspections.	More quality control procedures agreed upon and administered by the project team. It is needed as quality should be a given in Agile.
Requirements are challenging to identify and to have proper traceability to end product.	Recognise the importance of the Product Owner role and getting it right.
	Following agile-oriented requirement management process.
Requirements management with large/complex products is difficult.	Following agile-oriented requirement management process.
	Recognise the importance of the Product Owner role and getting it right.

	Conducting “Just Enough” (proper preparation) upfront work before the start of the Agile project (Design included - detail design in development).
Requirements priority lists are hard to create and maintain.	Following agile-oriented requirement management process.
	Recognise the importance of the Product Owner role and getting it right.
	Conducting “Just Enough” (proper preparation) upfront work before the start of the Agile project (Design included - detail design in development).
Requirements refinement are challenging.	Recognise the importance of the Product Owner role and getting it right.
	Following agile-oriented requirement management process.
Short, iterative and incremental deliveries are a challenge.	Educate on the value of regular delivery of software. Speaking of core Agile principles.
Specialised skills are needed; like refactoring, configuration management, developer unit testing (TDD), and various others.	Following agile-oriented configuration management process
	Implementing Test-Driven Development (TDD) and Continuous Integration (CI) are reasonable first steps. The tools for both of these practices are standard and (often) free.
	Motivate, educate and train teams in the use of Test-driven development (TDD), Pair programming, continuous integration (CI) and refactoring as starters.
	Companies should use tools that can supply incremental evolution, continuous integration, re-working, version management and support other agile practices.
Steep learning curve.	More critical is personal characteristics: Team members with attributes like collaborative attitude, honesty, sense of responsibility, eagerness to learn, and willingness to work with others are just as important as high skill level.
	Educate on the need of training investment and provide Agile and Method training for all teams and stakeholders, including new roles.
	In agile software engineering literature, the mature Agile coach has a critical role to play to ensure successful outcomes. An organisation needs to hire an Agile coach for at least six months to increases the chance of succeeding in the Agile adoption process
	Changing mindset takes time and managers should be continually mentored and coached.
	Mature Agile coaching and championing.
	Careful planning and discipline of agile adoption are required. A team must also expect a slight decrease in productivity when first adopting agile while the teams learn the new implementation techniques.
	Build a sharing and learning environment to empower team decision-making.
	Encourage task self-assignment to allow the developer to work in different areas and learn new skills.
	Use pair programming and pair rotation to distribute knowledge and facilitate learning.
Tailoring the appropriate Agile/Lean practices are difficult	Cherish Agile communities of practices.
	Ensure cross-team observation/validation of agile practices.
	Select appropriate personnel and provide the necessary training, mentoring and creating a set of work practices that promote process excellence.
	Focus on the XP core practice of whole team focus and educate on the value of it.
	Evaluating risks through an Agile assessment is the best overall approach to determining how much agility is enough in a transforming environment.
Team members are overly specialised, which requires work to be handed off several times.	To minimise hand-offs, encourage pairing at all stages and ensure that the necessary handoffs are informal. Encouraging the team to tackle small chunks of work also helps because it forces them to communicate more frequently.
Team members find it difficult to change the roles they are used to.	Integrating critical speciality skills to support Agile teams
	Implement change leadership and recognise the importance of it (Leadership owning the change initiative).
	Educate on the need of training investment and provide Agile and Method training for all teams and stakeholders, including new roles.
	Make the change transparent and open.
	Communicate the change intensively.
	Communicate that change is non-negotiable.

	Project manager plays the role of facilitator.
	Reintroduce specific roles when it is perceived beneficial to teams with, e.g. large team size, conflicts between developers.
	Recognise the importance of the Product Owner role and getting it right.
	In agile software engineering literature, the mature Agile coach has a critical role to play to ensure successful outcomes. An organisation needs to hire an Agile coach for at least six months to increases the chance of succeeding in the Agile adoption process
Team members, in general, find it difficult to work effectively in a team.	Every member of the team needs to contribute, and if one or two members of the team are putting themselves above the team, this needs to be quickly addressed by management. Often, this results in the quick removal of the individual(s).
Teams are committing to too much.	Committing to only what is sure and instead pull in more work when all work is done early.
Teams are split, and critical resources are removed after a successful pilot.	Emphasise value: the value of the team, and members of the team staying together and prioritised requirements.
Techniques and reporting in Agile to measure progress is difficult.	Good progress tracking mechanism that communicates well.
	When working with management, pay attention to both Agile metrics (or the lack, thereof), and adaptive planning over predictive planning. The Agile approach to both of these is counter-intuitive for most classically trained managers and requires constant reinforcing.
	Planning for specific technologies and tools are essential for collaboration, reporting and effectiveness.
	Invest in tools that support distributed agile teams.
The autonomous team model is challenging (self-organisation).	Explain the value of coherent, self-organising teamwork.
	Allow grassroots level empowerment.
The creation and estimation of user stories are hard.	Recognise the importance of the Product Owner role and getting it right.
	At the start of the project, it is good if various teams can agree on an estimation method and sizes for reference stories. E.g., Everyone understands what a user story size of three looks like. They all use it from there on as a reference. This way, all teams will have their velocity based on the same referencing system.
	Educate on the need of training investment and provide Agile and Method training for all teams and stakeholders, including new roles.
The customers can introduce changes as much as they want.	Allow the customer to have full authority.
	Excellent customer relationship and commitment (involvement), leading to higher customer satisfaction.
	In the situation where a single customer cannot be identified, pinpoint the project sponsor, who is the single person who ultimately approves the funding of the project. By working with the sponsor and clearly articulating the need for a single business representative on the project team, this situation can usually be quickly resolved.
	Customer company runs training sessions on essential Agile topics within the business domain and the company-specific area(s).
	Coaches should patiently coach the team on the value of the customer in the team.
	Let the teamwork directly with the customer
The focus is more on the individuals than the team (Over-individualism).	It is imperative that the whole team need to work well together. Every team requires a mature Scrum Master or Agile leader that will help to facilitate that. All team members' work should be evident.
The identification of real customer needs is difficult.	Excellent customer relationship and commitment (involvement), leading to higher customer satisfaction.
	In the situation where a single customer cannot be identified, pinpoint the project sponsor, who is the single person who ultimately approves the funding of the project. By working with the sponsor and clearly articulating the need for a single business representative on the project team, this situation can usually be quickly resolved.
	Customer company runs training sessions on essential Agile topics within the business domain and the company-specific area(s).

	Coaches should patiently coach the team on the value of the customer in the team.
	Let the teamwork directly with the customer
The implications of devolved decision-making (Team makes decisions)	Build a sharing and learning environment to empower team decision-making.
	Implement a democratic voting system.
	Project manager plays the role of facilitator.
The perceived gap between long and short-term planning	If multiple releases are followed, both long and short planning is done continuously. Otherwise, long-term planning is done in the initiation/inception/discovery phase.
	Mature Agile coaching and championing.
The perceived rewards for Using Agile Techniques are minimal.	Assess agility regarding agile values and principles, not practice adherence.
	Collecting and sharing successful adoption stories and positive experiences.
The role of a single identifiable Product Owner is missing.	In the situation where a single customer cannot be identified, pinpoint the project sponsor, who is the single person who ultimately approves the funding of the project. By working with the sponsor and clearly articulating the need for a single business representative on the project team, this situation can usually be quickly resolved.
	In the situation where several different groups have an equal interest in the success of a project, the team still needs a single representative who is willing to work with each of the different groups and prioritise accordingly. Again, work with the project sponsor to provide the path to a solution.
The Scrum Master refuses to protect the team.	If the Scrum Master is ineffective at protecting the team, this person needs to be replaced. To find a new Scrum Master, request volunteers or ask the team whom they would like to represent them.
There is a general organisational resistance to change	Utilising a maturity and risk assessment of the organisation will help to determine the main resistance criteria and how to scale. Tools like the AAC, 4 stage process (identifying discontinuing factors) and SAMI of Sidky (2007) might be useful.
	Evaluating risks through an Agile assessment is the best overall approach to determining how much agility is enough in a transforming environment.
	Establish guidelines for safe and agility-compatible process maturity assessments where ISO or CMMI cannot be followed.
	Engage everyone in the organisation.
	Identify incompatible assumptions (model clashes) and synergies between agile and traditional methods within your organisational processes.
There is a lack of change management during the adoption process	Communicate the change intensively.
	Make the change transparent and open.
	Implement change leadership and recognise the importance of it (Leadership owning the change initiative).
	Conduct empirical studies of which change are more unpredictable and therefore suited for agile and which are more predictable and suitable for traditional plan-driven methods.
	Upper management approval is fundamental to support any significant change in the processes of any department; without their consent, the success of the adoption is doubtful. To acquire the upper management support a presentation or a meeting might be conducted
	One of the valuable benefits of agile development is its ability to be customised based on the culture and the environment of the organisation it is adopted in. These changes should not compromise its beneficiary and affect its efficiency. It might not be
	Communicate that change is non-negotiable.
	Ensure management support.
	Make management support visible.
	Pragmatic customisation of the agile process
There is a need for agile-compliant performance evaluation	Performance evaluation needs to consider the breadth of skills, not just depth.
	Performance evaluation to apply much higher weighting for mentoring and voluntary contributions.
	360° feedback is necessary.

	Long-term solutions are dependent on rewarding teamwork and breadth of understanding. While there is no single or easy solution, some ideas include having team goals and bonuses, eliminating individual performance reviews, and allowing teams the opportunity to make mistakes.
There is scepticism towards the new way of working	Mature Agile coaching and championing.
	Assess agility regarding agile values and principles, not practice adherence.
	Try to have multiple 'bought-in' developers on each team.
	Changing mindset takes time and managers should be continually mentored and coached.
	A pilot project must be selected when first adopting agile, and it should be near the middle of what is the average for an organisation, small enough to be done by one team and should not be critical to the organisation
	In agile software engineering literature, the mature Agile coach has a critical role to play to ensure successful outcomes. An organisation needs to hire an Agile coach for at least six months to increases the chance of succeeding in the Agile adoption process.
	Agile teams must frequently communicate with senior management to clearly articulate their successes and challenges, and they should share their experiences with the broader organisation, so their achievements are acknowledged.
	Ensure management support.
	Make management support visible.
There is an increased reliance on social skills: e.g. Pair Programming and deeper need for collaboration.	Combine development and training program to provide customised training materials on social skills, using developers' examples.
	Using proper documentation to back up communication.
	Select appropriate personnel and provide the necessary training, mentoring and creating a set of work practices that promote process excellence.
There is a perceived need for developers to be a "master of all trades."	Use pair programming and pair rotation to distribute knowledge and facilitate learning.
	Encourage task self-assignment to allow the developer to work in different areas and learn new skills.
	Reintroduce specific roles when it is perceived beneficial to teams with, e.g. large team size, conflicts between developers
There is not a proper continuous integration or build system and process in place.	Implementing Test-Driven Development (TDD) and Continuous Integration (CI) are reasonable first steps. The tools for both of these practices are standard and (often) free.
Tracking velocity in a scaled environment is not clear.	Manage Effort Variance Constructively (issues causing a variance between different teams).
	At the start of the project, it is good if various teams can agree on an estimation method and sizes for reference stories. E.g., Everyone understands what a user story size of three looks like. They all use it from there on as a reference. This way, all teams will have their velocity based on the same referencing system.
The traditional need to overgenerate documentation. The perception that it is a safety net.	Documentation should be just enough. Define appropriate knowledge management strategy and distribution of knowledge in different levels of the organisation.
	Many organisations complain about the lack of documentation, but also over-documenting and spending plenty of time documenting unnecessary information is incorrect. What agile development offers is a new way of comprehensive documentation that takes less time and effort. This is another issue that needs to be supported by the upper management and agreed upon with the customers from the beginning of the project.
The uncertainty that Agile can scale appropriately for large projects.	Create and communicate positive experiences and celebrate small wins.
	A pilot project must be selected when first adopting agile, and it should be near the middle of what is the average for an organisation, small enough to be done by one team and should not be critical to the organisation
	Customise the agile approach carefully (small gradual improvements).

	Utilising a maturity and risk assessment of the organisation will help to determine the main resistance criteria and how to scale up. Tools like the AAC, 4 stage process (identifying discontinuing factors) and SAMI of Sidky (2007) might be useful.
	In agile software engineering literature, the mature Agile coach has a critical role to play to ensure successful outcomes. An organisation needs to hire an Agile coach for at least six months to increases the chance of succeeding in the Agile adoption process.
	Define specific functionality or responsibilities that you are going to address with agile approaches. (Specific Agile focus or large-scale adoption?)
	Reintroduce specific roles when it is perceived beneficial to teams with, e.g. large team size, conflicts between developers
Using non-flexible tools and hardware is a barrier to moving to agile.	Companies should use tools that can supply incremental evolution, continuous integration, re-working, version management and support other agile practices.
Variance on the speed of delivery between Agile and traditional running concurrently and having interdependencies.	While the whole team is not fully Agile, and there exist specific interdependencies between various method teams, key upfront agreements and risk acceptance have to be understood and agreed upon. This approach requires much discipline and takes away from the productivities a full Agile team could have achieved.
	Develop management and architectural practices for hybrid agile and plan-driven methods.
	Develop architectures that support the compartmentalisation of agile and traditional teams, where the hybrid approach is in place.

Table 2.29: Mapped possible solutions to challenges (taken from tables 2.24 and 2.28)

2.7 The Gap and Significance

From the information provided in the above review, it is clear that Agile adoption has many approaches and various frameworks available that can be applied in addition to these approaches. The four main approaches to adoption are:

1. Agile methodologies adoption: the focus is on one or more of the Agile methodologies and the manner in which to get these methods adopted by the various teams.
2. Agile practices adoption: the focus is on Agile practices, with the belief that if the relevant practices are instilled and followed, the organisation will become more Agile.
3. Agile adoption in an organisation: the focus is on the whole organisation and what it would take to achieve a cultural transition.
4. Agile adoption in teams: the focus is not on the broader organisation, but specifically on Agile adoption in those teams implementing the software. The belief is that if those teams understood the philosophy and bought into it, the methods and practices would be implemented easier.

This research focuses, as one of its constructs, on any of the approaches above, for the AAC acts as a tool to recognise possible challenges and to recommend possible solutions for these challenges. All of the approaches mentioned above have various frameworks and models to realise the outcome that they set out to achieve. There are no studies to date that present an exhaustive and exact mapping between challenges and solutions, and none of them offers an automated procedure to reach a potential list of possible Agile adoption solutions.

This review has repeatedly shown that the primary conceptual framework, as indicated in Figure 2.4, is unique and addresses the objectives as well as the problem statement with all its variables.

From the literature review, it is illustrated that no current work provides a comprehensive overview of the challenges, related solutions, critical trends in adoption challenges and prioritisation of all known Agile adoption challenges. No one existing Agile adoption framework was found to be suitable to feed into the formation of the online Agile adoption coach.

At the heart of the problem statement is the need to save time, improve the quality of adoption outcome and reduce the cost of the transition that is impacted. As an Agile coach would testify, it could take an inordinate amount of time in medium to complex systems to come to a reasonable conclusion as to what the probable solutions could be. The AAC has the potential to reach such a conclusion in a fraction of the time it usually requires and therefore speaks to a tremendous saving of both time and cost. The rigour introduced in putting the framework together will further add immense value towards the quality of the outcome concerning the completeness of findings.

2.8 Theoretical and Conceptual Framework

2.8.1 Theoretical framework

If one uses the metaphor of Grant and Osanloo (2014:16) in comparing a blueprint of a house to a theoretical framework, it is clear that even though this study's design is not very elaborate, its construction is rich. Burnard (2004:178) explains that it is not always necessary for a particular theoretical framework, as long as:

1. The researcher has sufficiently described what s/he was aiming to discover.
2. The researcher illustrates how data collection and analysis were performed.
3. The researcher indicates what was ultimately discovered through the research.

Sinclair (2007) compares a theoretical framework with a map or travel plan. One uses the experience and accounts of others who have been on the same trip to prepare and plan one's trip. Even though some have written on the main constructs depicted in the conceptual framework presented below, the existing theory does not provide a holistic overview or proper "travel plan" for all the constructs.

Kitson et al. (2008) argue that a single theory could limit a specific research's applicability, and they, therefore, suggest that a theoretical framework consists of multiple theories at different levels. Even if one follows this approach, the study will be strained as the individual "blueprints" for the main constructs, namely Agile adoption challenges, and solutions to these challenges are quite diverse. Furthermore, there lacks a clear theoretical framework that fits this study.

As can be seen in the conceptual framework below, the primary constructs are:

1. Agile adoption challenges.
2. Success factors and solutions for Agile adoption challenges.
3. Mapping Agile adoption challenges to success factors and solutions.
4. Verifying and testing automated Agile adoption coach.

Even though the existing theory describes the construct Agile adoption challenges (see (Dikert et al., 2016; Qumer & Henderson-Sellers, 2008 and Gandomani et al., 2013:622)) to a certain extent, it is not a well-trodden path or "blueprint" to be used as a research guide.

Turning to the second central construct, it again follows a similar path as Agile adoption challenges. Answers or solutions to Agile adoption challenges can be found in literature in different places. In some studies, it speaks directly to "success factors" (including (Misra et al., 2009; Nguyen, 2016:175-179; Brown, 2013:122 and Chow & Cao, 2008:963)), thereby focusing on what it ought to be and not necessarily from where one is coming and how one could change it. The majority of other studies incidentally provide

solutions, but no definitive research indicates a full set of solutions mapped to challenges (see (Conboy et al., 2010:49; Nerur et al., 2005 and Hajjdiab & Taleb, 2011a)). Soundararajan and Arthur (2011) highlight the goodness of Agile, which is yet another position on what could constitute solutions for specific Agile adoption challenges. It is clear therefore that this construct does not have sufficient history, clarity and depth to be viewed as a proper theory or blueprint to be used (according to the premise of Kitson et al. (2008) described above).

The third and fourth constructs are unique and were deduced from constructs one and two with their pertaining independent variables (indicated in the conceptual framework). As it is a *deduction* from existing constructs, there is no base for an existing theoretical framework on constructs three and four either.

This study, in conclusion, cannot make use of any specific formal theoretical framework, but would use the outputs (independent variables) from various studies (based on the first two constructs) and analyse, structure and deduce certain vital extrapolations which would then result in the formation of the adoption framework (mapping challenges to various possible solutions). This adoption framework will be automated and have specific dependent variables associated with it. It will further make use of the three principles from Burnard (2004) described above as guiding themes, together with the conceptual framework discussed below.

This research follows a simple sequence of events.

> literature review → analysis of the data → survey on priority → analysis of priority data → structuring of data in an Agile adoption framework → implementation using the AAC → survey on AAC → analysis of AAC survey

2.8.2 **Conceptual framework**

Grant and Osanloo (2014:16-17) describe a conceptual framework as highlighting the researcher's view on the best manner the problem can be explored, the overall direction of the research as well as how the variables are interrelated in the study.

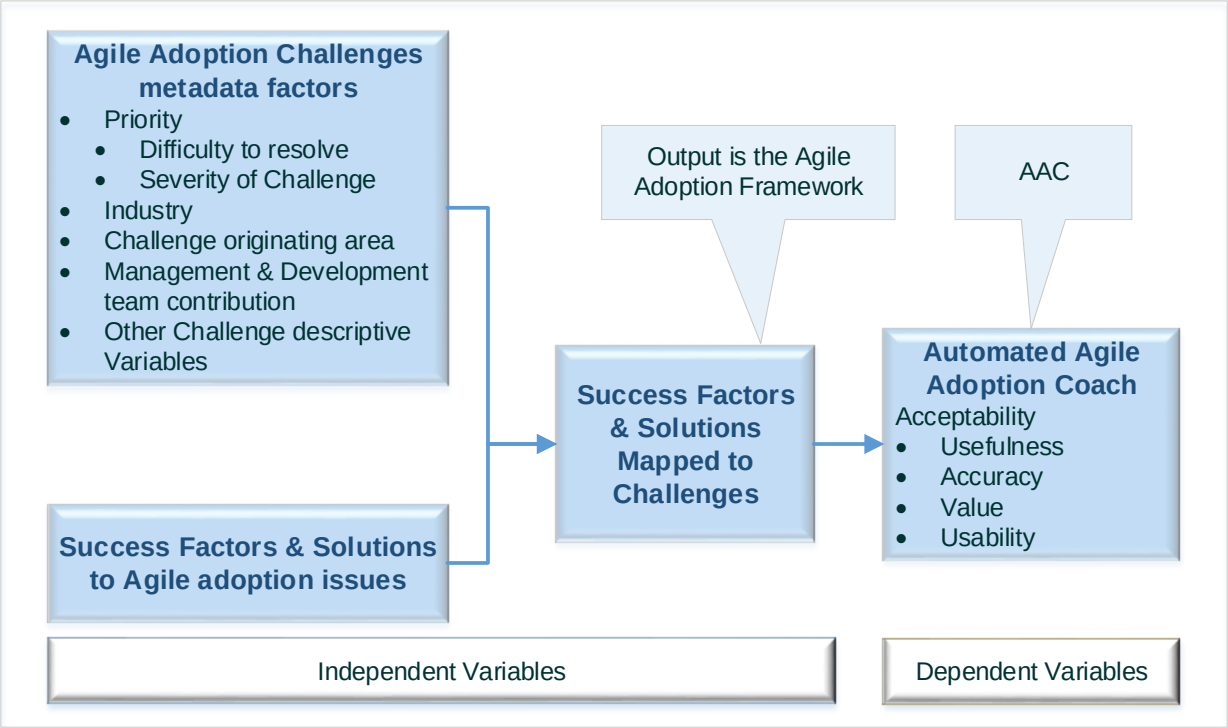


Figure 2.4: Conceptual framework of the study

The above sequential conceptual framework acts as a holistic overview of the research, including all the main constructs and variables.

2.8.3 **Constructs**

Besides the main constructs mentioned above, and speaking about the relationship they have with one another, the various variables are also clearly indicated. The relationship between the first two constructs is elaborated on in the literature review. The third construct, namely the Agile adoption framework, is the critical input and engine to the AAC. The last construct (automated Agile adoption coach) has four dependent variables attached to it, which will be discussed below. Besides the variables, there are two further critical activities associated with the AAC construct, namely to build the AAC and secondly to have it assessed by a sample group of Agile coaches and Scrum Masters.

2.8.4 **Independent Variables**

A change in independent variables can, in turn, lead to a change in the dependent variables (Saunders, Lewis & Thornhill, 2016). In light of this, there are two types of

independent variables. The one type (**group 1**), by their existence or non-existence, will impact the dependent variables. This group contains three of our main constructs (challenges, solutions and their mapping). If any of the items in these constructs are missing, for example, a specific challenge, a solution or incorrect mapping, it will impact the dependent variables.

The second type (**group 2**) includes those independent variables that will help the study to ask the correct questions in the AAC. The primary outcome (constituting the Agile Adoption Coach) requires us to know what questions to ask the user to come to a satisfactory result. These questions ought to help us narrow the field and come to a more precise and prioritised answer concerning what possible solutions to focus on. The independent variables below ought to help us shape these questions.

Independent variables:

Group 1

- Challenges
- Success factors & solutions
- The mapping between challenges & solutions

Group 2

- Priority of a challenge (difficulty and severity inform the priority)
    - Difficulty in resolving
    - The severity of a challenge
- Industry
- Challenge originating area
- Management & development team contribution
- Other challenge descriptive variables

All of the variables above will be studied separately, and additionally, specific variables will be examined concerning others (e.g. priority needs inputs from severity and difficulty). As indicated in the conceptual framework, the severity, together with difficulty, will constitute the priority value of a challenge.

2.8.4.1 Group 1: Challenges, Success Factors & Solutions

Challenges, success factors and solutions have been discussed in detail in the literature review section. Challenges as a construct are furthermore the leading independent variable. Without it, or if the extrapolation from the literature is incorrect, the AAC will be nullified. The need to find comprehensive normalised lists of challenges, success factors and solutions forms the basis of this study.

Similarly, the success factors and solutions identified in the literature review as a construct are an indispensable part of the AAC. If critical solutions are not identified, the value of the AAC will suffer. The AAC is inherently not only concerned with the challenges but finding solutions according to the priority of the challenges.

2.8.4.2 Group 1: Mapping between Challenges & Solutions

The mapping of the challenges correctly to the various potential solutions identified is what brings the AAC together. Without mapping or incorrect mapping, the AAC will not be acceptable. The normalised list of challenges and solutions could add some value for Agile coaches, but it is the mapping that will set it apart.

The Agile adoption framework is deduced from this mapping and is a structured database which takes input from all of the independent variables in groups one and two

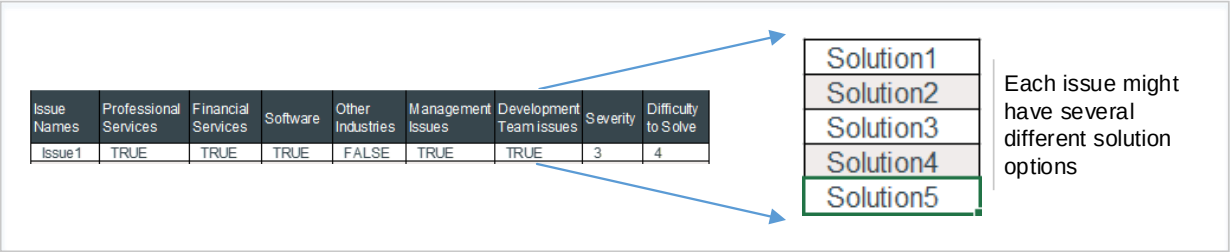


Figure 2.5: Conceptual view of how a challenge can map to various possible solutions

2.8.4.3 Group 2: Priority (Difficulty & Severity)

Dikert et al. (2016:48) emphasise the importance to ascertain which challenges are more critical so priority can be established. This view is of considerable significance as it speaks about the priority of the challenges and how it will be displayed in the AAC to the user. These two factors operate in direct opposition to one another. If one challenge is of great importance to resolve, for example, but at the same time extremely difficult to solve, it

might be more prudent to focus on another important issue which is not that difficult to solve first. To prioritise these challenges, it is essential to assign a weight to the impact of challenges as well as the difficulty. The formula and the manner in which to use it to deduce the priority can be found under section 3.7.5 under the main heading “Research Design and Methodology”.

This manner eases the extrapolation of a prioritised list of challenges to focus on, which could assist the Agile coach in obtaining a few quick wins in practice.

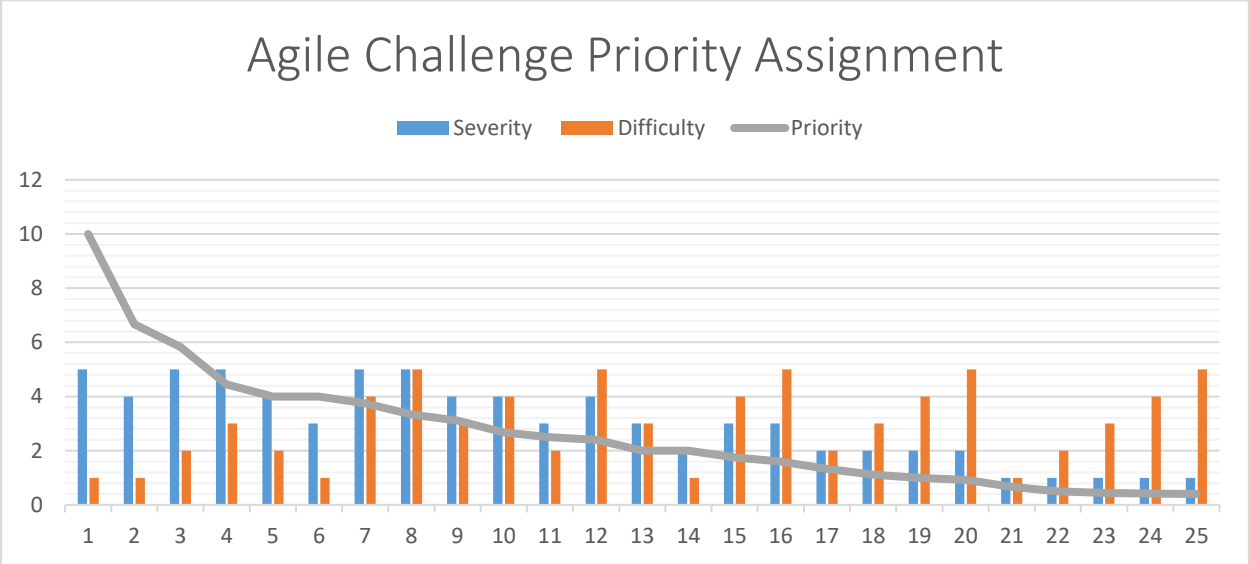


Figure 2.6: Priority Assignment graph concerning Challenges (See also **Table 3.14**)

Figure 2.6 above is a conceptual representation of the relationship severity and difficulty has with priority. The vertical bars indicate the weighted average of an issue and the horizontal the different associations a challenge might have concerning priority and difficulty.

2.8.4.4 Group 2: Industry

This variable relates to how Agile adoption challenges are distributed among various industries. Conforto et al. (2014:21) indicate that there are indeed different challenges in different industries, and proposes a hybrid Agile management system for various industries.

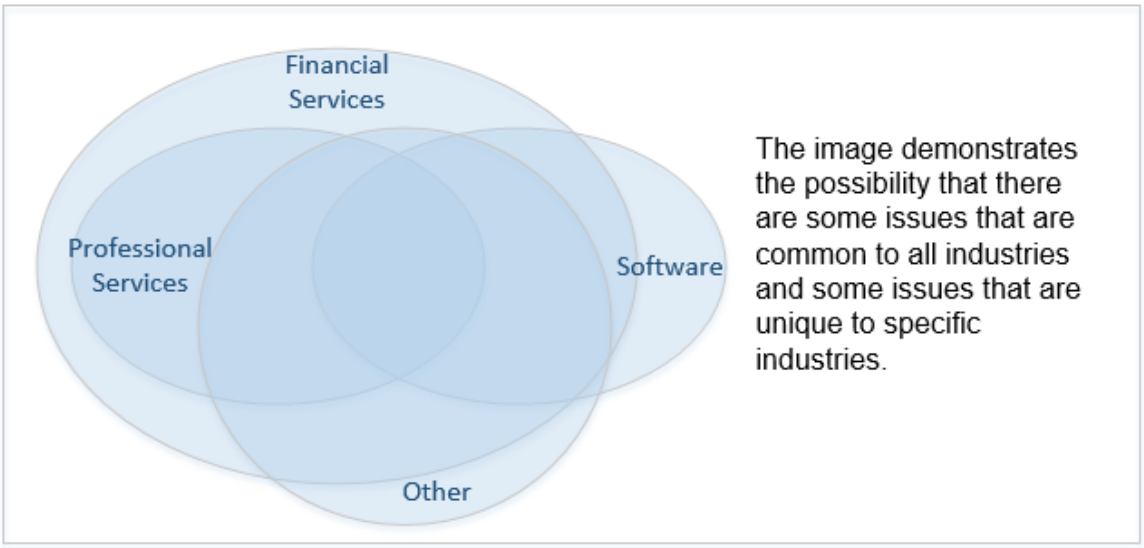


Figure 2.7: Possible distribution of Agile challenges among the three key industries

These overlapping domains' conceptual framework in Figure 2.7 illustrates how the challenges unique to industries could help in narrowing the field of possible solutions in the AAC; however, the challenges common to all industries are those that all coaches ought to place an extra emphasis upon to understand and know how to approach.

Actual finding: In the conclusion section for 2.4.4, the study found that Agile adoption challenges are for the most part similar across industries, or that there is not enough research to conclude whether there is a definitive mapping between industries and challenges. The hypothesis that **industries** might be an independent variable was found to be **invalid.**

This independent variable has been selected as a topic for potential further study; however, as it is inconclusive for this study, it is not factored in as a truly independent variable.

2.8.4.5 Group 2: Challenge Originating Area

Challenge originating areas were found throughout the literature review and are indicated in Table 2.21 under section 2.4.3. These challenge originating areas are mutually exclusive, and the distribution is illustrated in Figure 2.8 below.

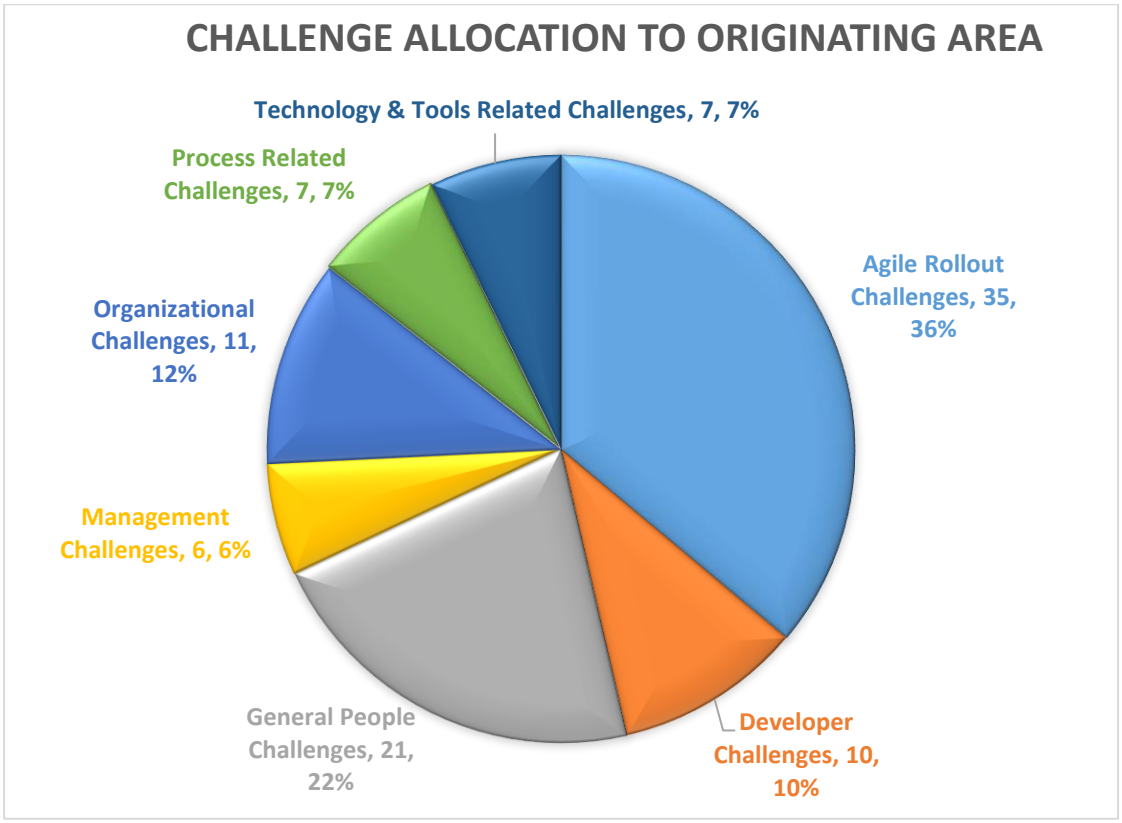


Figure 2.8: Challenges distribution through Originating areas

2.8.4.6 Group 2: Management and Development team's Contribution to Challenges

Conboy et al. (2011:2) expands a great deal on the Agile value "people over processes" and focus specifically on the management layer impact on Agile adoption. Alternatively, Adkins (2010:21) presents a key focus using Agile coaches to identify, facilitate and resolve Agile challenges introduced by the development team.

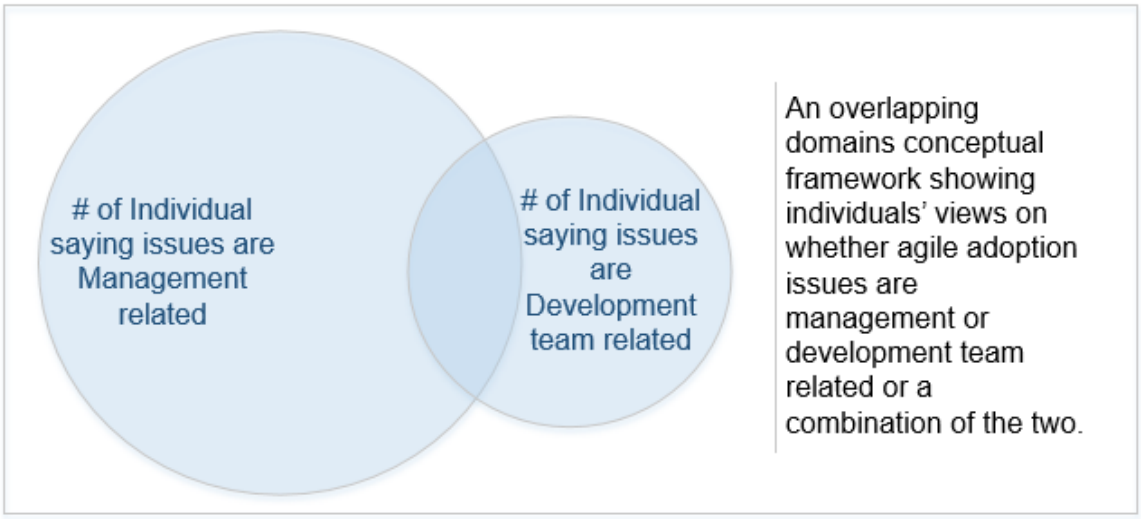


Figure 2.9: Possible views on management and development team challenges

The overlapping domains conceptual framework in Figure 2.9 above illustrates that a significant portion of individuals indicate that agile adoption challenges are management-related, whereas a much smaller section indicates that it is development team-related. There are further some people who suggest that it is a combination of the two. As a user selection-point in the AAC, this option could potentially also assist in narrowing the field of outcomes.

Actual finding: The literature review found that management and developer-related challenges are indeed relevant to the AAC, but not as stand-alone individual variables. As indicated in Figure 2.8 above, management and developer challenges are incorporated into the challenge originating areas.

2.8.4.7 Group 2: Other challenge descriptive variables

The assumption is that there are variables that are unknown, and the purpose of this sub-objective is to define what those variables could be so that the AAC questionnaire may be enriched further.

This hypothesis was studied in depth concerning variables to the challenges. An additional independent variable was found in general challenge categories, as stipulated by Dikert, Paasivaara and Lassenius (2016). The detail concerning this variable can be

found in section 2.4.3.2. This list of categories is further mutually exclusive inside this general list of categories and is illustrated in Figure 2.10 below.

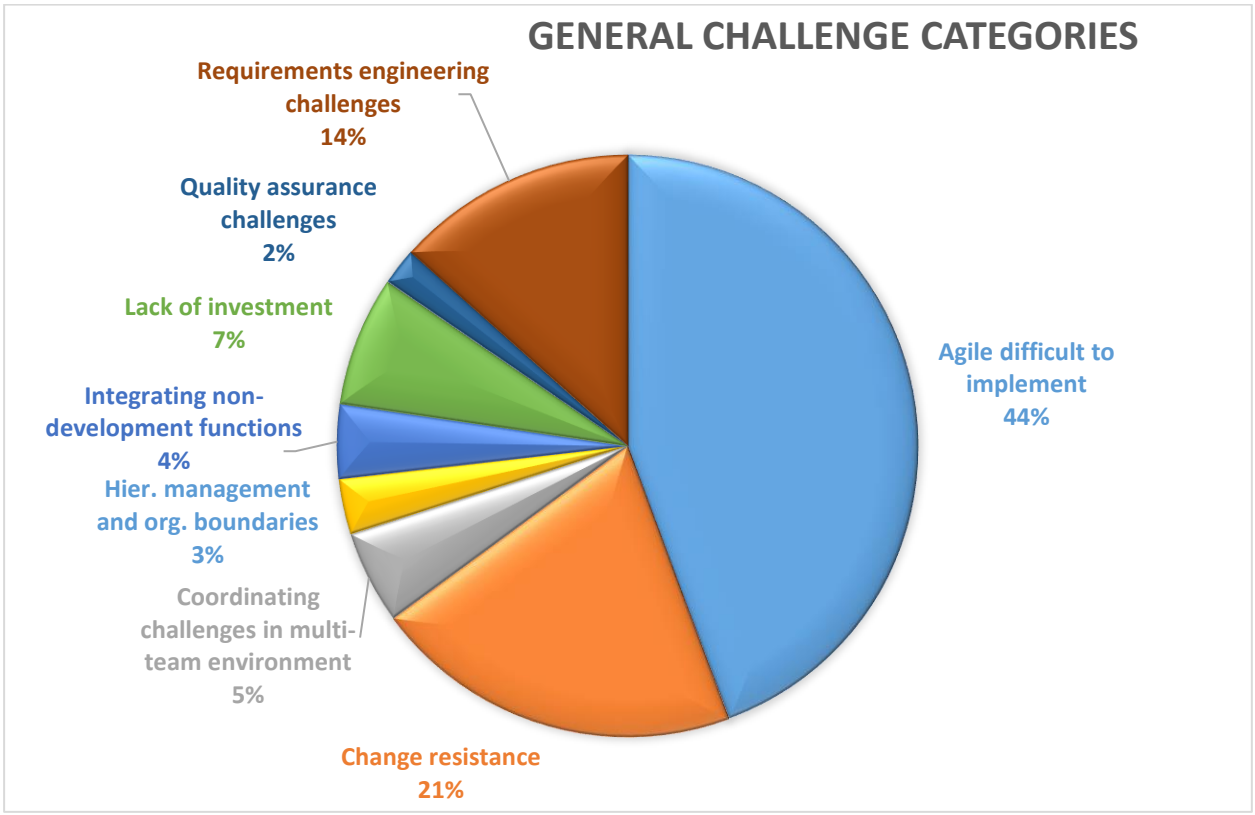


Figure 2.10: Pie chart showing challenges distributed in broad categories

This independent variable will further be built into the AAC as a way of positioning the questions in a structured manner to the users.

2.8.5 **Dependent Variables**

The following dependent variables are the primary variables that have to be measured to determine the actual success of the framework, as well as answering the central research question. CAMP (2011:34) argues that dependent variables speak to the outcomes of interest, and Saunders, Lewis and Thornhill (2016:444) expand on this notion by stating that dependent variables change in respect to inputs from other variables.

The dependent variables can only be measured by the individuals (Agile coaches and Scrum Masters) who will potentially use the system. It will be a subjective view; however, with a sample group larger than 30 Agile coaches and Scrum Masters, the expectation is

that the ratings will support the hypothesis that the AAC can be an acceptable tool for Agile coaches.

According to CAMP (2011:34), interval variables have distances between the attributes while having ratio properties. All of the interval-dependent variables below have a scale between one and five, where five denotes total agreement and one disagreement.

1. **Usefulness**: The AAC needs to be useful. Merriam-Webster (2018) defines useful as “capable of being put to use; especially serviceable for an end or purpose”. If the Agile coaching community does not think that the AAC is capable of being put to use, the AAC has failed in its purpose.
2. **Accuracy**: Accuracy has an extreme dependency on the independent variables relating to challenges, solutions and their mapping. If any of these are compromised, accuracy will be impacted. The higher the perceived accuracy, the more acceptable the AAC will be.
3. **Value**: Value speaks mostly to the definition of Merriam-Webster (2018) relating to either the relative worth, utility, or importance of the AAC, but it can also speak to the fact that the AAC can save organisations money. Locating prioritised challenges and their mapped solutions faster can definitively add to the financial savings a company can have in utilising this tool. All of the problem statement items in Table 1.1 are addressed in this dependent variable.
4. **Usability**: Usability in general software development tend to be neglected (Speicher, Both & Gaedke, 2015); however, it is one of the critical determinants in keeping the user engaged and interested in the interaction with the user interface. If this is not well-rounded, acceptance of the AAC will be impacted.

The variables above will be populated from the feedback information requested from the user at the end of the AAC. If the user has agreed to do this and has completed the AAC, he or she will be asked four questions as presented in Table 3.12. These four dependent variables culminate in the understanding of whether the AAC will be acceptable for Agile coaches or not. All of these variables will be tested thoroughly through statistical analysis.

An additional question will be asked in the pilot phase to determine whether any improvements are needed:

- Are there any improvements that you would like to suggest to this system?

The results of this question will be worked back in the AAC to provide a better-tested AAC and user experience. If any of the independent variables change, the primary dependent variables “Acceptability” will also be impacted. It would be challenging to statistically test the relationship between the dependent and the independent variables. The reasons are:

1. The method to produce the output on the literature review on Agile adoption challenges and success factors was quite rigorous. It is not expected that these independent variables will change rapidly.

2. The survey to deduce the priority on Agile adoption challenges was quite onerous and time-consuming. Feedback on the AAC will be monitored with respect to the priority but to perform subsequent surveys to deduce a new priority is quite costly. It is therefore also not expected that this variable will change often.

3. The variables on challenge categories and originating areas have been taken from literature, and there is not an expectation that this will change soon either.

It is a definitive fact that should any of the above independent variables change the output of the dependent variable will also be impacted. If however, changes in points 1 and 2 above are accepted by the Agile community, then the acceptability of the dependent variable should not change. Should changes be introduced in 1 and 2 above without consideration from the Agile community, the acceptability would obviously be impacted as well. The use of the AAC actually depends on the independent variables to stay relatively constant. The dependent variables are further expanded upon in section “Phase 3 (AAC acceptability) Data analysis”.

2.9 Summary

This chapter is kicked off with a discussion on the value of Agile adoption and how a tool that can streamline the process can be significant. A comparison was made with other

Agile adoption frameworks to see if there is uniqueness in the proposed tool. The conclusion is that this framework has a very particular focus in that it supports the first phase of the adoption process and could be used as a helper tool in any other holistic Agile adoption framework to speed up the adoption process as well as to assist in the integrity of the outcomes.

This literature review chapter encompasses much more than a review of the available literature on the subject discussed. A specific quantitative survey was done on the literature to determine an extensive database of Agile adoption challenges (97), together with a normalised database of possible Agile adoption solutions and success factors (142). Furthermore, a particular mapping between Agile adoption challenges and possible challenges was established based mostly on literature and some from the author's own experience (mapping to 13 challenges).

The following five of the research sub-questions are answered in this chapter, specifically in sections 2.4 and 2.5:

a. What are the general, unique lists of Agile adoption challenges and success factors in literature?
b. What is the relationship between the various Agile adoption challenges originating areas?
c. Are all industries involved in software development similarly impacted by Agile adoption challenges?
d. What are the other variables relevant to Agile adoption challenges in Agile software development? The answer to this question will assist in building a comprehensive filter for the automated Agile adoption coach.
e. What is the mapping between Agile adoption challenges and recorded possible solutions to these challenges in the Agile software development space?

Finally, the theoretical and conceptual framework is discussed together with the related constructs, dependent and independent variables, represented in Figure 2.4.

The research design and methodology chapter which follows describes the approach of the research together with the various instruments used.

The last research sub-question on Agile challenges priority and the main research question on whether the AAC can be acceptable in the Agile coaching community is addressed in chapters four and five, respectively.

Chapter 3: RESEARCH DESIGN AND METHODOLOGY

3.1 Introduction

According to MacMillan and Schumacher (2001:166), Leedy and Omrod (2013:96) and Terre Blanche, Durrheim and Painter (2007:23) a research design establishes the overarching logical plan and framework so that it can serve as a bridge to answer the research questions. Leedy and Omrod (2013:96) go as far as to say “Research design is planning”.

Generally, research methodology describes the “how” in how an enquiry should proceed in reaching the desired conclusion of the research question (Creswell, 2003; and Shareen, 2010:39). Leedy and Omrod (2013:76) expound on this by asserting that research methodology describes the techniques and instruments one uses to collect and analyse data.

Together, the research methodology and research design, as part of this chapter, aims to provide sufficient information to reproduce the study if any researcher so wishes (Grademiners, 2015).

The previous chapter, the literature review, played a prominent role in assisting in generating the database for the AAC. This database came partly to fulfilment by combining and normalising the quantitative results of multiple studies on challenges and solutions. Chapter 3 follows the research design “onion” metaphor as stipulated by Saunders, Lewis and Thornhill (2016:124). The onion’s layers are considered in this study from the outer layer (the philosophy) towards the sixth layer (practical techniques and procedures) in the centre.

The research design approach results in three different quantitative studies performed in this research. These studies are represented in three distinct chronological phases: phase 1 (Challenges & Solutions), phase 2 (Challenges Prioritisation) and phase 3 (AAC acceptability)) shown as sub-headings under this chapter. These phases are elaborated on in Table 1.2.

This chapter includes an overview of the philosophical lens adopted in this study, an approach to theory development, exploration of the methodology and the strategies alluding to that. Also, Chapter 3 presents the time frame, the techniques and procedures to do data collection and data analysis for the research and ends with a discussion on the validity and reliability of the research.

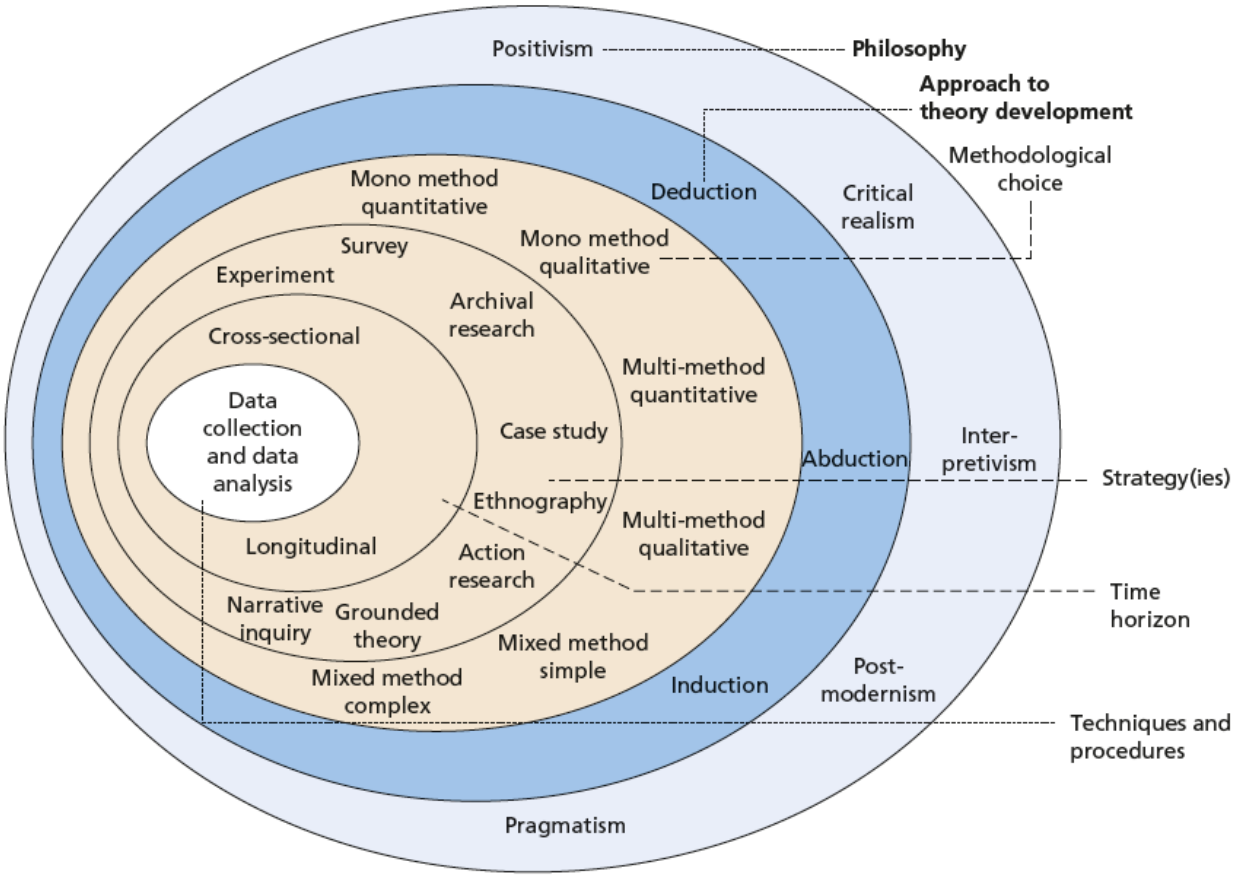


Figure 3.1: Research Design "onion" as presented by Saunders, Lewis and Thornhill (2016:124)

3.2 Research philosophy

Saunders, Lewis and Thornhill (2016:135) mention the following five essential philosophies in business and management research: positivism, critical realism, interpretivism, postmodernism and pragmatism. Table 3.1 below attempts to provide a short overview of these philosophies and how they are related to ontology, epistemology and axiology as well as the typical methods used with each approach. After each approach, a conclusion is reached with respect to the value of the approach for this study.

Creswell (2003:6,25) briefly refers to ontology as the claims that researchers make about what knowledge is, as well as describing the nature of reality. Saunders, Lewis and Thornhill (2016:127) agree with this view and go further to state that ontology determines the way one sees the world, and would, therefore, influence the type of research which could be selected.

Epistemology concerns itself with how we perceive knowledge, what is acceptable and valid, and how one would communicate this knowledge to an audience (Burrell & Morgan, 1979:3). Oates (2006) additionally asserts that epistemology is about *how* one acquires knowledge.

Axiology concerns what values go into knowledge (Creswell, 2003:6) and how one deals with one's values and those of one's research participants (Saunders, Lewis & Thornhill, 2016:128).

3.2.1 **Philosophical perspective of this research**

The conclusion reached in Table 3.1 below is that positivism is the best approach for all the phases of this study.

Ontology	**Epistemology**	**Axiology**	**Typical methods**
Critical realism			
Stratified/layered (the empirical, the actual and the real) External, independent Intransient Objective structures Causal mechanisms	Epistemological relativism Knowledge historically situated and transient Facts are social constructions A historical causal explanation as contribution	Value-laden research The researcher acknowledges bias by worldviews, cultural experience and upbringing The researcher tries to minimise bias and errors The researcher is as objective as possible	Retroductive, in-depth, historically situated analysis of pre-existing structures and emerging agency. The range of methods and data types to fit the subject matter
Conclusion: This approach is disqualified as an option because a researcher acknowledges bias in his or her worldview and this study cannot allow bias in any way.			
Interpretivism			
Complex, rich Socially constructed through culture and language Multiple meanings, interpretations, realities The flux of processes, experiences, practices	Theories and concepts too simplistic Focus on narratives, stories, perceptions and interpretations New understandings and worldviews as a contribution	Value-bound research Researchers are part of what is researched, subjective Researcher interpretations key to the contribution Researcher reflexive	Typically, inductive. Small samples, in-depth investigations, qualitative methods of analysis, but a range of data can be interpreted

Conclusion: This approach is disqualified due to its intrinsic qualitative nature, and for this study, all three phases take a quantitative approach.			
Postmodernism			
Nominal Complex, rich Socially constructed through power relations Some meanings, interpretations, realities are dominated and silenced by others The flux of processes, experiences, practices	What counts as 'truth' and 'knowledge' is decided by dominant ideologies Focus on absences, silences and oppressed/repressed meanings, interpretations and voices Exposure to power relations and challenge of dominant views as a contribution	Value-constituted research Researcher and research embedded in power relations Some research narratives are repressed and silenced at the expense of others Researcher radically reflexive	Typically, deconstructive – reading texts and realities against themselves In-depth investigations of anomalies, silences and absences Range of data types, typically qualitative methods of analysis
Conclusion: The deep qualitative nature of this approach and the power relations involved disqualifies postmodernism as an approach for this study.			
Pragmatism			
Complex, rich, external 'Reality' is the practical consequences of ideas The flux of processes, experiences and practices	The practical meaning of knowledge in specific contexts 'True' theories and knowledge are those that enable successful action Focus on problems, practices and relevance Problem-solving and informed future practice as a contribution	Value-driven research Research initiated and sustained by the researcher's doubts and beliefs Researcher reflexive	Following the research problem and research question The range of methods: mixed, multiple, qualitative, quantitative, action research Emphasis on practical solutions and outcomes
Conclusion: In general pragmatism comes close as an approach candidate, but as it is value-bound and reflexive as opposed to value-free and detached, it is also disqualified.			
Positivism			
Real, external, independent One true reality (universalism) Granular (things) Ordered	Scientific method Observable and measurable facts Law-like generalisations Numbers Causal explanation and prediction as contribution	Value-free research The researcher is detached, neutral and independent of what is researched Researcher maintains an objective stance	Typically, deductive, highly structured, large samples, measurement, usually quantitative methods of analysis, but a range of data can be analysed
Conclusion: This approach is the closest match to all three phases of research in this study and is explained further in the two paragraphs below.			

Table 3.1: Comparison of five research philosophies (Saunders, Lewis & Thornhill, 2016)

Positivism stipulates hypotheses or research questions that are testable and allows for conclusions which are measurable. These measurements are typically done against accepted variables, and knowledge of the world one lives in (Saunders, Lewis & Thornhill, 2016:137). Saunders, Lewis and Thornhill, (2016:137) clarify further that a researcher's own beliefs have no value to influence research, as he or she is neutral and independent

towards the research study. Easterby-Smith et al. (2012) posit that the positivism philosophical approach is mainly related to the observations and experiments to collect numeric data and relates mostly to quantitative research approaches (such as this study follows).

The ontology clearly lends itself to an ordered list of challenges and solutions, which is the primary outcome of this phase, hence supporting the positivist approach. Clearly, measurable facts with respect to the challenges and solutions can be observed in literature as part of the epistemology. As the researcher is detached and only relay back the findings on challenges and solutions found in literature, the axiology also supports the positivist approach.

The research under this phase explores the theory in a quantifiable and objective manner and relates to the importance that was posited already in existing research, hence positivism (Wikipedia, 2018). The researcher stood wholly free and detached from the value in the research. The sample size is, however, limited due to the amount of formal research already completed on Agile challenges and success factors. The philosophical approach to Agile challenges and success factors are the same. It is only the specific focus areas in Agile, which is different.

A quantitative questionnaire was sent out to Agile coaches and Scrum Masters to establish the prioritisation of the items after a list of Agile challenges and success factors was obtained from the literature review. The reason why Scrum Masters were also included is that as part of their role, the coaching aspect plays quite a significant part. Typically coaches would be in an excellent position to ascertain their companies' challenges prioritization. The research under this phase has a very singular outcome: To determine the priority for every Agile challenge stipulated. It makes use of a survey with only two questions for every Agile challenge. Both these questions are bounded to the Likert-scale. A large sample size, consisting of both Scrum Masters and Agile coaches, was used for this survey. The researcher was detached and objective in this scientific and quantifiable approach. This qualifies the research in phase two as a positivist philosophical approach as well. The emphasis in this phase was to ascertain quantifiable results that lend themselves to statistical analysis.

In order to ascertain the usefulness, accuracy, usability and contribution of the AAC (incorporating the challenges and success factors) to the Agile field, another survey was conducted - again amongst Agile coaches and Scrum Masters. The research under this phase furthermore makes use of a survey, albeit a more complex one than in the prioritisation survey, as it consists of four questions to determine the usefulness, accuracy, usability and contribution to the Agile field. It also makes use of the five-point Likert-scale. In respect to the sample size, it is similar to the previous phase as the sample group consists of both Agile coaches and Scrum Masters. A combination of scientific methods, referenced in section 3.7.6 (data analysis on the AAC acceptability), were used to measure and analyse the results and is repeatable in any future study. The researcher did not influence those participants involved in the survey, and the researcher maintained an objective stance throughout the study. Based on the deductive and highly structured approach, this phase can also be seen as conforming to positivist philosophy.

3.3 Methodological choice

Under methodological choice, it is essential to understand the nature of quantitative and qualitative approaches as well as the hybrid nature of a mixed qualitative and quantitative approach. The next step is to make a selection of whether to follow a quantitative, qualitative or mixed-methods research design (see Creswell (2003:3), and Saunders, Lewis and Thornhill (2016:164)).

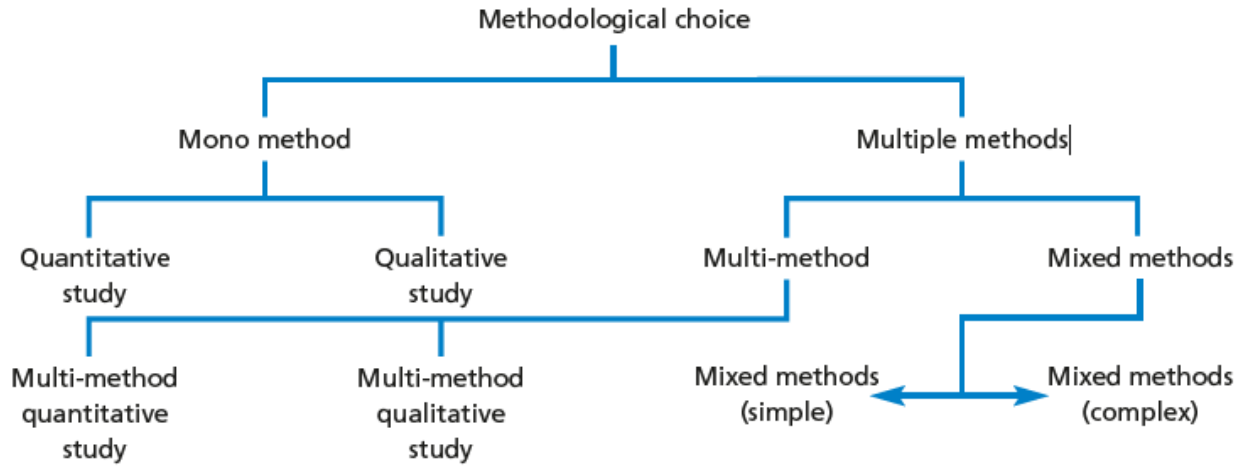


Figure 3.2: Methodological choices (Saunders, Lewis & Thornhill, 2016:167)

Quantitative research focuses on verifying hypotheses (deductive) or finding patterns (inductive), typically making use of large amounts of data (Hinkelmann & Witschel, 2013:15). Abawi (2008:3) states that quantitative research is a process of inquiry-based on testing a theory composed of variables, measured with numbers and analysed using statistical techniques. Cohen and Manion (1980) highlight that quantitative research is social research that employs empirical methods and empirical statements. Dawson (2008:15) practically explains quantitative methods as research that generates statistics using large-scale survey research and utilising techniques like questionnaires or structured interviews.

Dawson (2008:14) argues that qualitative research is focussed on interaction with research participants to explore their attitudes, behaviour and experiences. The methods used are generally focus groups and interviews. The end goal is to gain a deeper understanding of the participants' opinions. Hinkelmann and Witschel (2013:15) succinctly state that qualitative research focuses on understanding the essential characteristics of typically small samples of data.

Figure 3.2 above provides a hierarchical breakdown of methodological choices. A researcher typically has a choice between a mono-method (selecting either a quantitative or qualitative research design) or multi-method (a multiplicity of similar methods or a mix of the two).

The approach in this research is to follow a multi-method quantitative research design. Referring to Table 1.2, the phases allow for a logical progression from one phase to the next. In effect, phase 1 is intermingled with phase 2. This is a result of both these phases having the aim of producing the AAC database. Phase 2 allows for completing the AAC database in deducing the priority of challenges. Researchers from various fields of study, recommend the use of a multi-method approach to study complex social phenomena (Saunders, Lewis & Thornhill, 2016:166; Creswell, 2003), as it is likely to overcome weaknesses associated with using only a single method (Bryman, 2006:106). Bryman (2006:98) asserts that the multi-method approach is beneficial towards a more productive approach to data collection, analysis and interpretation.

3.4 Approach to Theory Development

Conventionally, three modes of theory development exist: deductive, inductive and abductive (Saunders, Lewis & Thornhill, 2016).

A deductive approach follows from the general to the specific. The typical flow in deductive theory development is first to develop a theory from experience, observations or previous findings. A particular hypothesis is secondly derived from that theory. Observations are thirdly made from the analysis of the data, from which the confirmation or rejection of the hypothesis can follow as a fourth and final step (Hinkelmann & Witschel, 2013:13). To simplify, deductive theory development uses a top-down approach to draw conclusions based on the general (Mouton, 2005).

An inductive approach is almost the polar opposite from a deductive on as it flows from the specific to the general. Analysis and observations are the first steps, after which specific patterns are extrapolated. Understanding the patterns, the researcher formulates a hypothesis which s/he can explore and validate. Finally, a theory can be formed from these hypotheses (Hinkelmann & Witschel, 2013:13). In the inductive approach, theory generation, rather than theory verification or falsification, is vital (Saunders, Lewis & Thornhill, 2016).

Abductive theory development makes use of known premises, and these are used to create testable conclusions. Generalisation happens in the interactions between the specific and general. From themes and patterns, a conceptual framework is generated. This framework, however, needs to be tested and validated further by subsequent data and -analysis. It either aims to generate or modify existing theory (Saunders, Lewis & Thornhill, 2016:145). Instead of moving from the general to the specific (as in deduction) or the specific to the general (as in induction), an abductive approach moves backwards and forwards, combining the deductive and inductive approaches (Suddaby, 2006:639).

The mode of theory development in phases one and two follow an inductive approach matched to the description above. A theory is only developed at the end of these phases. Phase 3, however, follows a deductive approach based on the need for theory verification or falsification.

3.5 Research Strategy

Saunders, Lewis and Thornhill (2016:178) describe eight research strategies or methods in the fourth layer of their research onion, which a researcher can use.

- Experiment - testing hypotheses
- Survey - finding patterns in data
- Archival and documentary research – using online archives and documents for research
- Case study - studying the characteristics of a real-life instance
- Ethnography - studying the culture or social world of a group
- Action research - iteratively solving a problem with a community of practice
- Grounded theory – both a method of inquiry and the result of a research process
- Narrative inquiry - using a story or a personal account which interprets an event

3.5.1 Phase 1 (Challenges & Solutions)

This phase involved putting together an extensive list of agile adoption challenges and success factors obtained through a literature survey as detailed in section 2.3. The next logical, independent variable and phase following the generation of these normalised lists are to determine the priority of the challenges.

3.5.2 Phase 2 (Challenges Prioritisation)

As previously indicated (Methodological choices), the combined output of phases one and two delivers the AAC database. The output of phase 2, however, provides the final overview of the priority of the Agile Adoption challenges. It satisfies the second sub-objective: "… identify the priority of the various Agile adoption challenges using a survey." Phase 2 makes use of a survey using a questionnaire. Where phase 1 makes use of a literature review as a method, phase 2 focuses on a questionnaire as a method to gather data. Both phases, however, follow the survey strategy.

The hypothesis for phase 2 on establishing a priority for challenges is:

If Agile coaches and scrum masters are surveyed on the difficulty and severity of Agile adoption challenges, we should be able to formulate a prioritized list of Agile adoption challenges.

To understand the priority of Agile adoption challenges, it is critical to understand both the difficulty and severity of the impact of a particular Agile adoption challenge. A large sample group, consisting of Agile coaches and Scrum Masters, answering the two questions regarding difficulty and severity provides sufficient input for ascertaining an acceptable outcome in respect to the priority of the Agile challenges.

An online survey determines the priority of the challenges. SmartSurvey (2018) mentions the following (Table 3.2) 10 advantages of online surveys.

1. Faster	6. Easy to use for researchers
2. Cheaper	7. Easy to style
3. More accurate	8. Honesty
4. Quick to analyse	9. More selective
5. Easy to use for participants	10. More flexible

Table 3.2: Online survey benefits (SmartSurvey, 2018)

All 10 of these advantages are relevant to the survey questionnaire in phase 2, specifically in comparison with paper questionnaires, which have significant challenges for large sample groups. Some of these challenges are: slow to return rate, higher expenses, rigorous analyses and less flexibility (Milne, 1999; Sukamolson, 2005).

3.5.3 **Phase 3 (AAC acceptability)**

The strategy choice in phase 3 is similarly that of a survey - making use of an online questionnaire technique. The strategy outcome in Phase 3 provides the critical answer to the primary research question: “Will an automated agile adoption framework be accurate, acceptable and useful to Agile coaches, helping companies adopting Agile software development?”

Phase 3 follows a deductive approach concerning the hypotheses (whether the research question postulated is valid or not). To establish whether the AAC tool is acceptable,

accurate and usable requires a quantitative survey strategy to measure specific responses to the questionnaire given to the sample group.

Babbie and Mouton (2012:263) assert that another advantage of survey questionnaires is that they ask the same question for every participant, and therefore, the threat of bias is negated.

In summary, the strategy for the various phases is represented in Table 3.3 below:

Phase	Strategy	Method
1	Survey	Literature Research
2	Survey	Questionnaire
3	Survey	Questionnaire

Table 3.3: Strategies for the Phases 1 - 3

3.6 Time Horizon

According to Saunders, Lewis and Thornhill (2016), the question the researchers have to ask themselves, is whether the research needs to represent a snapshot in time (cross-sectional study) or where snapshot entries need to be made over time (longitudinal study). The answer to this will provide a particular time horizon.

Saunders, Lewis and Thornhill (2016:200) indicate that cross-sectional studies mostly implement the survey strategy as it needs to investigate a specific phenomenon in a particular time. Saunders, Lewis and Thornhill (2016:200) continue to assert that most studies follow the cross-sectional study approach due to time and financial constraints.

As all three of the research phases make use of the survey strategy to research specific phenomenon at a given snapshot in time, all three of the phases is also using the cross-sectional time horizon.

3.7 Techniques and Procedures

3.7.1 Phase 1 (Challenges & Solutions) Data collection

The literature reviewed for the study is a distribution of formal (academically published) literature research outcomes as well Internet resources in the form of forums, blogs, and websites with a critical focus on Agile adoption challenges and success factors. The motivation for the latter being that the Agile community is quite active, with Agile coaches publishing and engaging online much more frequently than within formal literature.

Scopus, an online academic article and books database, and Google Scholar were the primary databases accessed for searching relevant literature in this field. On 10 October 2016, Scopus provided 334 results for the search term “Agile adoption”. Refining the search by including the word “challenges” it resulted in 177 results. Out of these results, together with that of Google Scholar, a selection of material that speaks widely to Agile adoption challenges and success factors was made. Fifteen academic publications were included for challenges and twenty-four for success factors. Additionally, a selection of two success factors and four informal Internet resources for challenges were made to devise a comprehensive list of Agile adoption challenges and success factors. The same search terms as for the formal literature above were used.

Quality formal lists for challenges and success factors were hard to find, but the formal literature sample for both these data types was sufficient to provide a comprehensive list of challenges and success factors. Very little informal material was subsequently needed. With the quality and credibility of some of the online platforms being questionable, this dissertation focused on online sources and data from well-known and acknowledged coaches alone. On 10 October 2016, a significant number (47 500) of hits were found on Google for the search term [“Agile adoption” challenges]. On the same day, 16 000 hits were returned on Google for the search term [“Agile adoption” success factors]. A large percentage were of poor quality, and due to the excellent quality of the formal literature, only a few of these sources were selected to be included. They are highlighted in Table 3.4 and Table 3.5 below. Table 3.5 additionally indicates which studies have direct mappings to challenges. These informal resources defined the challenges and success factors clearly and came from reputable sources and Agile coaches.

Referenced Resource	Type	Data Field
(Conboy et al., 2010)	Journal	Challenges

(Dikert, Paasivaara & Lassenius, 2016)	Journal	Challenges
(Gandomani et al., 2013)	Journal	Challenges
(Vijayasarathy & Turk, 2008)	Journal	Challenges
(Nerur, Mahapatra & Mangalaraj, 2005)	Journal	Challenges
(Noruwana & Tanner, 2012)	Journal	Challenges
(VersionOne, 2016)	Website	Challenges
(Hajjdiab & Taleb, 2011a:8-9)	Journal	Challenges
(Srinivasan & Lundqvist, 2009)	Conference	Challenges
(Rodríguez et al., 2012)	Conference	Challenges
(Hajjdiab & Taleb, 2011b)	Conference	Challenges
(Pitkänen, 2015)	Dissertation	Challenges
(Mishra & Mishra, 2011:551)	Journal	Challenges
(Boehm & Turner, 2005)	Journal	Challenges
(Scrumology, 2009)	Website	Challenges
(Almeida, 2017)	Journal	Challenges
(Marchenko & Abrahamsson, 2008)	Conference	Challenges

Table 3.4: Formal and informal literature for Challenges

Referenced Resource	Type	Success Factor (SF) / Solution
(Dikert, Paasivaara & Lassenius, 2016)	Journal	SF / Solution – **Mapped**
(Conboy et al., 2010)	Journal	Solution - **Mapped**
(Scrumology, 2009)	Website	Solution - **Mapped**
(Pitkänen, 2015)	Dissertation	SF
(Boehm & Turner, 2005)	Journal	SF / Solution – **Mapped**
(Almeida, 2017)	Journal	Solution
(Hajjdiab & Taleb, 2011b)	Conference	Solution - **Mapped**
(Marchenko & Abrahamsson, 2008)	Conference	Solution
(Sidky, 2007)	Dissertation	Solution
(Brown, 2013)	Book	SF
(Gandomani et al., 2013)	Journal	Solution - **Mapped**
(Nguyen, 2016)	Journal	SF
(Misra, Kumar & Kumar, 2009)	Journal	SF
(Chow & Cao, 2008)	Journal	SF
(O'Connor & Duchonova, 2014)	Journal	Solution
(Nivoit, 2013)	Dissertation	Solution
(Bavani, 2009)	Conference	SF
(Kakar, 2017)	Journal	Solution
(Lindstrom & Jeffries, 2004)	Journal	Solution
(Ramesh, Cao & Baskerville, 2010)	Journal	Solution
(Gorans & Kruchten, 2016)	Conference	SF

(Fitzgerald & Stol, 2014)	Journal	Solution
(Kropp, Martin & Meier, 2015)	Journal	SF
(De Haaff, 2017)	Website	Solution
(Anderson, 2003)	Book	Solution
(Schwaber, Laganza & D'Silva, 2007)	Website	Solution
(Sintes, 2017)	Website	Solution
(Beck et al., 2001)	Website	Solution
(Cohn, 2005)	Book	Solution

Table 3.5: Formal and informal Literature for Success Factors

A significant contributing factor towards quality data resulted from formal literature. Some of these publications reflect a limited study on challenges (Hajjdiab & Taleb, 2011a; Dikert, Paasivaara & Lassenius, 2016; Almeida, 2017), as well as success factors (Chow & Cao, 2008; Misra, Kumar & Kumar, 2009; Pitkänen, 2015; Dikert, Paasivaara & Lassenius, 2016; Nguyen, 2016).

Leedy and Omrod (2013:207) mention two sampling design types: probability and non-probability sampling. Each of these supports multiple sampling techniques.

In probability sampling, elements are randomly selected from the overall population, and it is assumed that the characteristics of the sample represent the characteristics of the larger population (Saunders, Lewis & Thornhill, 2016:275). Additionally, any part of the population has the potential to be represented in the sample (Leedy & Omrod, 2013:207). According to Saunders, Lewis and Thornhill (2016:276), probability sampling is mostly used in surveys.

Saunders, Lewis and Thornhill (2016:276) posit four stages in the probability sampling approach.

1. Identify a suitable sampling frame based on your research question(s) and objectives.
2. Decide on suitable sample size.
3. Select the most appropriate sampling technique and select the sample.
4. Check that the sample is representative of the target population.

Table 3.6: Four stages of probability sampling (Saunders, Lewis & Thornhill, 2016)

In non-probability sampling, there is no manner of predicting that each characteristic of the larger population will be presented in a random sample, with some members of the population having slim or no chance of being sampled (Leedy & Omrod, 2013:214).

In light of the statement of Leedy and Omrod (2013:214) on non-probability sampling, Phase 1 has followed a non-probability sampling approach with a purposive sampling technique. Patton (2002:45) highlights that purposive sampling requires the selection of specific information-rich samples. Saunders, Lewis and Thornhill (2016:301) emphasise that purposive sampling is usually used with small samples, as well as when the researcher wishes to select samples that are particularly revealing. In this research to select material on a probability sampling approach would have yielded very low-quality research material and this research wanted to show the challenges and solutions from the best studies available at the time and therefore a very purposeful approach.

Filtering through hundreds of studies and articles took an inordinate amount of time as part of the purposive data collection phase, in order to land up with the studies reference in tables 3.4 and 3.5.

3.7.2 **Phase 2 (Challenges Prioritisation) Data collection**

If it is impossible to ascertain data from every possible instance of the data, known as a census, the researcher needs to consider following a sampling approach (Saunders, Lewis & Thornhill, 2016:272). In a sampling approach, only a sub-group is selected, and they represent the entire population to a certain extent. Saunders, Lewis and Thornhill (ibid.) posit that a census does not always provide a better result than a distinct and selected sample group, and further assert sampling is a valid alternative where significant time and budget constraints exist.

Phase 2 uses non-probability sampling.

In phase 2, sampling is carried out by an online questionnaire tool. The focus is on scarce sources such as Scrum Masters and Agile coaches, and the fact that the sampling frame is impossible to determine, the non-probability sampling approach is used during this phase

3.7.2.1 Sampling Frame

The sample frame speaks to the complete list of all the elements in the population from which the sample will be taken (Saunders, Lewis & Thornhill, 2016:277). A sample frame constituting the whole Agile coaching and Scrum Master community globally would be impossible to determine. The sample frame size is considerable, and globally, there are no complete database or databases available for all Agile coaches and Scrum Masters, subsequently leaving a sampling frame unavailable. It is further unconducive to narrow the sample size to achieve a possible sample frame as the study's focus is to deliver a valuable AAC tool that can be used by all Agile coaches and not only by a selected few in a specific country or group.

3.7.2.2 Sampling Size

Probability Sampling Size

In general, according to statistics, the probability of errors goes down, the more significant the sample size grows and hence comes closer to the normal distribution. This relationship is called the central limit theorem (Saunders, Lewis & Thornhill, 2016:711). According to Saunders, Lewis and Thornhill (2016:279), there are four factors to consider when selecting a specific sample size. They are:

1. The confidence in the data collected.
2. The tolerable margin of error.
3. The number of categories the data will be sub-divided into.
4. The proportion of the target population.

Tennent (2013) remarks that there should not be a sample size of less than 30 for each category in probability sampling as the mean distribution below 30 statistically becomes more error-prone. Black (2010) emphasises that researchers require certainty of 95% or more for the sample size to represent the characteristics of the target population. The sample size might otherwise be error-prone.

Phase 2 recognises only one category despite both Agile coaches and Scrum Masters falling in this category. The behaviour concerning the recognition of difficulties and

severity of impact to the organisation of these two roles are identical as both these roles earnestly focus on removing impediments and Agile challenges from the organisation.

Non-Probability Sampling Size

According to Saunders, Lewis and Thornhill (2016:297), non-probability sampling (or non-random sampling) is ambiguous, and there is, in fact, no rules about sizing. They go on to explain that the logical relationship between the sampling technique and the focus of the research is essential. A key separating factor between probability and non-probability sampling size is that non-probability sample sizing makes generalisations to theory, rather than about a populace.

The sample group in phase 2 consists of Scrum Masters and Agile coaches. These respondents hail from various groups and sites. The following online forums and websites in Table 3.7 were proposed to canvas for respondents to complete the questionnaire:

1. Deloitte global Agile community of practice
2. Twitter
3. Scrum user group of South Africa
4. Scrum.org
5. LinkedIn Agile communities design

Table 3.7: Sample areas for phase 2 and 3 respondents

The study aimed for the response of at least 30 people to the various methods of canvassing (Table 3.7). The study achieved to solicit 51 respondents to this survey to indicate severity and difficulty for each Agile challenge. This number of people provide a sufficient normal distribution for the priority of the various challenges. Tennent (2013) and Saunders, Lewis and Thornhill (2016:280) indicated that the central limit theorem could occur even if the population is not normally distributed. Saunders, Lewis and Thornhill (2016:280) observe that statisticians have shown “that a sample size of 30 or more will usually result in a sampling distribution for the mean that is very close to a normal distribution.” This is why a result of 51 respondents was deemed sufficient. The fact that most of the respondents to the priority questionnaire came from the South African context is not problematic for the following reasons.

1. The study was meant to pull respondents from diverse countries, but it did not aim to categorize respondents from various countries.

2. Agile adoption challenges impact humans across the globe, and the priority on challenges should be very similar. E.g. lack of training will have a significant impact everywhere and can be seen as a relatively quick win and therefore, will have a high priority everywhere as well.

3. The output of the priority assessment is not seen as a final result but rather as an initial stab at a priority for various challenges. We know that Agile and companies are evolving and so would the challenges priorities as well. The AAC will update the challenges priorities as more feedback is received from people across the globe.

4. Most of the respondents were from one country, namely South Africa, and therefore, the outcome of the survey can still be applied to the South African context.

3.7.2.3 Sampling technique

The probability sampling techniques are “simple random”, “systematic random”, “stratified random”, cluster and multi-stage.

Table 3.8 represents the non-probability sampling forms concerning groups and techniques:

Group	**Technique**
Quota Purposive	Quota Extreme case Heterogeneous Homogeneous Critical case Typical case

	Theoretical
Volunteer	Snowball Self-selection
Haphazard	Convenience

Table 3.8: Non-probability sampling forms (Saunders, Lewis & Thornhill, 2016)

The sampling design technique in phase 2 follows a non-probability volunteer approach incorporating both **snowball** and **self-selection techniques**. Corbucci et al. (2011:5) used snowball and convenience techniques together as they incorporate online questionnaires, and it was impossible for them to ascertain a specific frame sample. A volunteer approach was selected, as no random selections were made, but rather particular people volunteered to fill in the questionnaire. Following the orange line in the decision tree in Figure 3.3 below, it clarifies why the volunteer technique was selected. There is no way of determining a complete sample frame in an online questionnaire where Agile coaches and Scrum Masters are canvassed from various online forums and websites to fill in the questionnaire for Agile challenges prioritisation.

In the self-selection technique, the chances that the sample is representative is low, and it is typically used where access to respondents is challenging while the research is more exploratory. Similarly, the snowball sample representation is small; however, the cases have a higher likelihood of the desired characteristics. It is typically used where respondents are difficult to be identified (Saunders, Lewis & Thornhill, 2016:298).

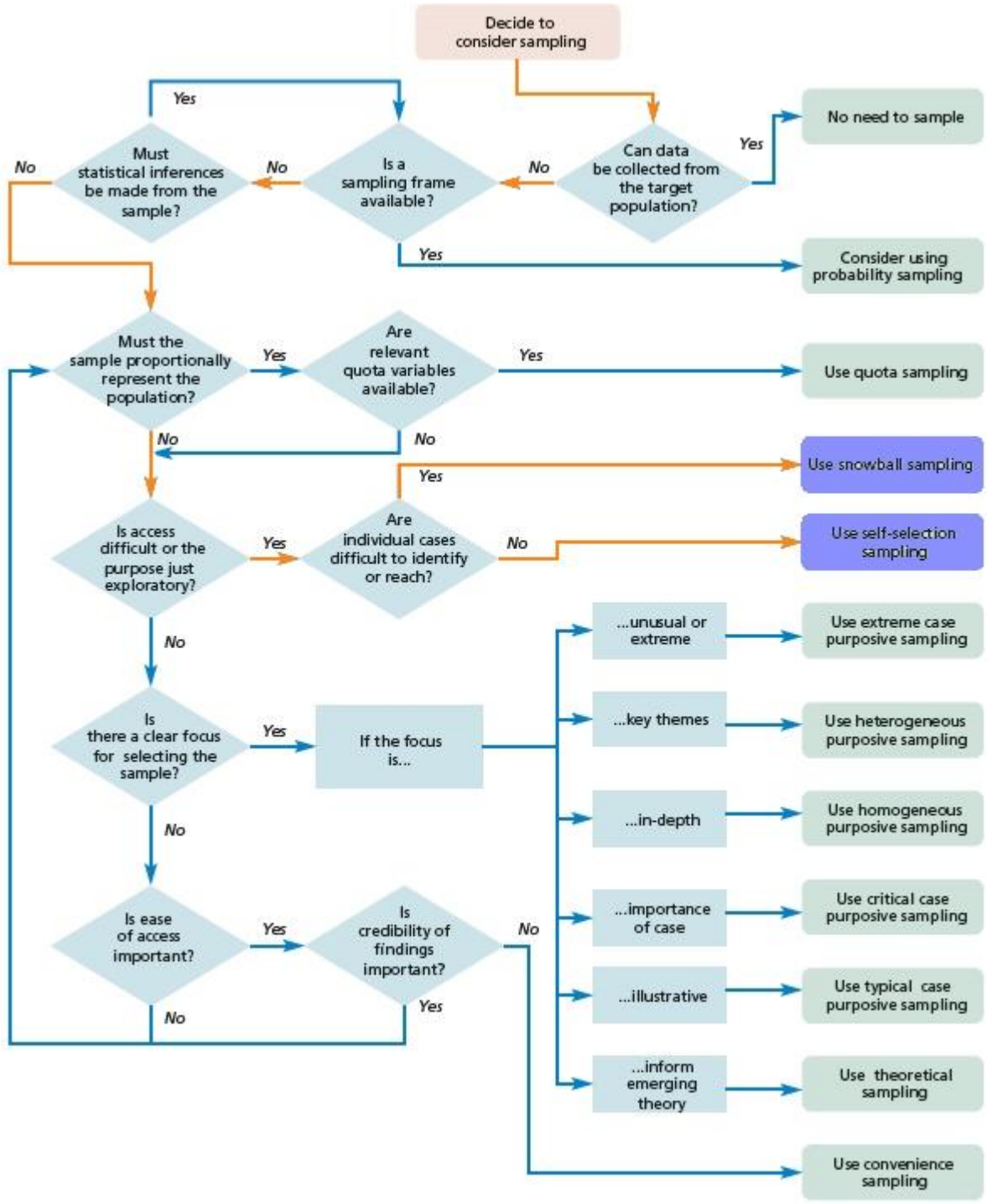


Figure 3.3: Decision tree for volunteer sampling (Saunders, Lewis & Thornhill, 2016:296)

3.7.2.4 Instrument

Data collection, as part of phase 2, was done using the questionnaire instrument. The primary purpose is to ascertain the priority of challenges so that the AAC database could

be completed, and then serve to display to the end-user an adequately ordered list of activities in respect to the priority of the challenges.

Milne (1999) lists three advantages of questionnaires:

1. The responses are gathered in a standardised way, so questionnaires are more objective, indeed more so than interviews.

2. Generally, it is relatively quick to collect information using a questionnaire. However, in some situations, they can take a long time not only to design but also to apply and analyse (see disadvantages for more information).

3. Potentially information can be collected from a significant portion of a group. This potential is not often realised, as returns from questionnaires are usually low. However, return rates can be dramatically improved if the questionnaire is delivered and responded to during class time.

Table 3.9: Advantages of questionnaires (Milne, 1999)

All three of these advantages apply to the online questionnaire in phase 2, primarily since the method uses an online survey questionnaire and not a physical one.

There are only two variables per challenge which the users had to complete. The respondents needed to indicate the difficulty in resolving a particular challenge as well as the severity (the impact on the business). Each of these variables follows the Likert-scale and is presented in Table 3.10 below. Each of the parameters (difficulty and severity) uses their Likert-scale to make the user referencing as comfortable as possible.

Likert number	Difficulty	Severity
1.	No difficulty at all in resolving the challenge.	No noticeable impact on the business
2.	Somewhat difficult to resolve.	Minimal impact on business
3.	Difficult to resolve the challenge	Noticeable impact on business
4.	Very difficult to resolve the challenge	Big impact on business

5.	Extremely difficult to resolve	Major impact on business

Table 3.10: Likert scale number legend for difficulty and severity of challenges

3.7.3 Phase 3 (AAC acceptability) Data collection

As with phase 2, phase 3 also required a sampling approach as the survey requires the views from the Agile coach community. It would be impossible to include the entire Agile coach population in the data gathering process.

The sampling design type in phase 3 also followed a non-probability design approach as in phase 2, as the non-random sample group of Agile coaches and Scrum Masters who needed to be solicited via the online questionnaire had a small chance of representing the whole Agile coaching community. The hope was that to increase the sample size, the normal distribution of the findings would become less error-prone.

3.7.3.1 Sampling Frame

Phase 3 aimed to formulate a highly probable theory that the AAC would be valuable, accurate and usable to the Agile coaching community. No database could help set a sample frame for Agile coaches globally. A sample frame was therefore not possible in phase 3.

3.7.3.2 Sampling Size

A sampling group of 30 or more Agile coaches and Scrum Masters were determined to be sufficient to extrapolate a view on whether the AAC assessment tool would be valuable for the Agile coaching community or not. Agile coaches, in general, are quite scarce. The solicitation of Agile coaches to use the AAC tool and give feedback on the questionnaire below was therefore done internationally. The sample group of Agile coaches and Scrum Masters came from the same resources list mentioned in Table 3.7 above. There were eventually 42 valid responses to the AAC acceptability survey.

3.7.3.3 Sampling technique

The sampling techniques were the same as in phase 2, where the snowball and self-selection techniques were listed.

3.7.3.4 Instrument

As stated in Table 3.3, phase 3 also made use of the survey strategy, making use of an online questionnaire instrument. The questions that were presented to the respondents aimed to ascertain whether the AAC tool would be accurate, usable, user-friendly and valuable to the Agile coaching community. The five-point Likert-scale, shown in Table 3.11 below, was used to grade the responses.

Likert number	Possibilities
1.	Never
2.	Seldom
3.	Sometimes
4.	Often
5.	Almost always

Table 3.11: A Likert-scale for questions in phase 3

The various questions that were presented to the respondents in phase 3 are summarised in Table 3.12 below:

Number	Questions
1.	Did you find the AAC user interface intuitive and easy to navigate?
2.	Would you recommend the AAC free assessment tool to other Agile coaches?
3.	Were the results of the assessment tool in line with what you expected (accuracy) it to be?
4.	Were the results of the AAC assessment tool useful to you?

Table 3.12: Questions posed to respondents in Phase 3 (AAC acceptability)

3.7.4 Phase 1 (Challenges & Solutions) Data analysis

Saunders, Lewis and Thornhill (2016:183-184) assert that primary and secondary data analysis are the two main types of data analysis. Phase 1 utilised a secondary data analysis approach as part of the literature review. Secondary data analysis includes an analysis of existing formal literature for a specific purpose. Saunders, Lewis and Thornhill

(2016:183-184) declare that to render desirable and acceptable outcomes from such an approach; the secondary data needs to be sufficiently trusted.

Arppe (2008) specifies three data analysis types:

- Univariate – uses one variable
- Bivariate – uses two variables
- Multivariate – uses three or more variables

The quantitative studies on the challenges and the success factors were done together from a bivariate perspective as only one variable in both cases (determining a normalised list of challenges and success factors respectively) was of concern, however, the associations between these variables needed to be considered to produce the mapping amongst each another.

Further to the types of data analysis, analysis can also be divided into descriptive and inferential categories. Descriptive analysis is concerned with the data the researcher has in hand. The inferential analysis involves making inferences about the findings beyond the immediate data in the research (Cyfar, 2018). Both variables used descriptive analysis on the quantitative studies as we were only interested in getting normalised lists of data (challenges and success factors) that can represent an acceptable "good enough" list of items and then produce a mapping between the variables. The intention of phase 1 is not to infer anything from the data.

Following the orange line in Figure 3.4 below the reasons for the selection of the data analysis type in phase 1 is described.

"Can the data be measured numerically as quantities?" → As mentioned above the data analysis takes a bivariate approach; therefore, two main categories (challenges and success factors) immediately come into play.

Agile adoption challenges from the literature also present specific categories and originating areas. These groupings are not definitive groupings qualified by rigorous quantitative research, but instead adopting the grouping names from some authors

(Gandomani et al., 2013; Dikert, Paasivaara & Lassenius, 2016) in the literature and then adding challenges to these grouping in a subjective manner. Saunders, Lewis and Thornhill (2016:500) state that categorical data should be unambiguous and discrete; as this assertion cannot be applied to the categories and originating areas of Agile challenges, they are not considered as definitive categories. Besides the authors of the literature study who have assigned challenges to these groupings, further subjective assignments have been made to divide all the remaining challenges.

Various graphs (Figure 2.2 and Figure 2.3) were produced to indicate the originating areas and categories of challenges.

The answer to the question stated in Figure 3.4, “Can the data be classified into more than two sets?” is “No”. Only challenges and success factors were considered, as well as the relationships that exist between them.

In conclusion, it is clear that phase 1 follows a descriptive (Dichotomous) data analysis approach.

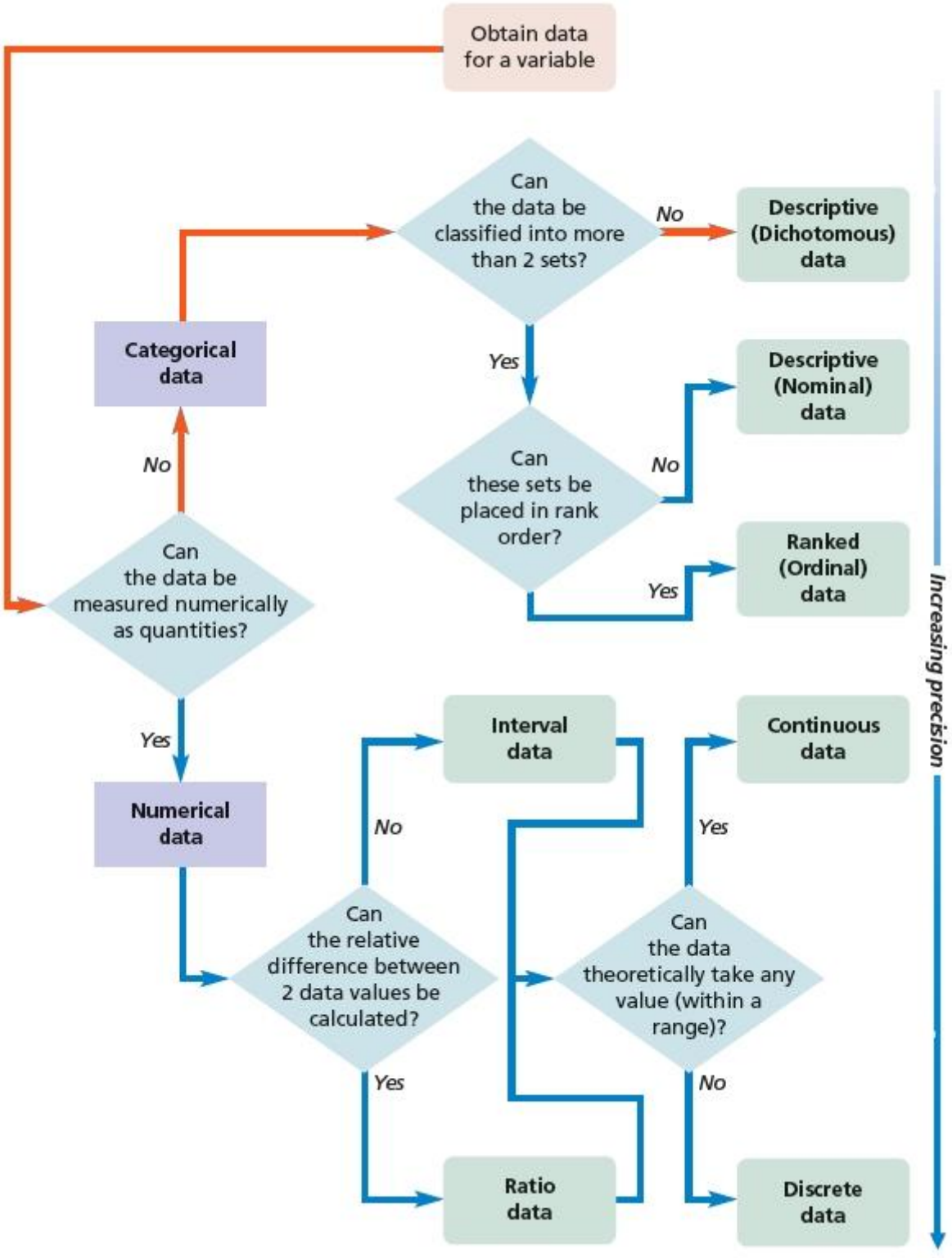


Figure 3.4: Data type selection path (Saunders, Lewis & Thornhill, 2016:499)

All the challenges and success factor data were added to a Microsoft Access database linking it to one another whenever the literature indicated so. Linking between challenges and possible solutions/success factors was further done objectively by the author, and part subjectively on originating areas and Agile challenge categories.

It is essential that no redundancies exist in either the challenges or success factors lists and that the linking to originating areas and categories are established. When all challenges and success factors were added from the literature lists in Table 3.4 and Table 3.5, a manual clean-up exercise commenced ensuring that no redundancies exist on either list.

Working from the top of the list, this redundancy clean-up followed a four-step process:

1. Use keywords in challenge or success factors to search the rest of the list. E.g., the search term “training” will be used for the challenge “Lack of training” in the rest of the list.

2. If redundancy is found, check for links to referenced literature and relink to the first finding of the challenge or success factor. E.g., another “Lack of training.” challenge was found. Find the reference for this challenge and link it to the first finding of this challenge.

3. Delete redundancy.

4. If no more redundancies are found for a particular challenge, start with a synonym search on every entry until no more redundancies are found. E.g., the search term “enthusiastic” was used in the challenge “Certain teams and individuals are overzealous”.

Table 3.13: Four-step process to normalise challenges and success factors and solutions

Two normalised lists were ultimately created for challenges and success factors/solutions, respectively.

As explained in section 2.6, the bivariate output of the challenges and success factors/solutions allows the formulation of the mapping between challenges and solutions.

A Microsoft Access database was used to store the data for these two variables. A simple SQL query was used to create the output between these tables.

3.7.5 **Phase 2 (Challenges Prioritisation) Data analysis**

As part of the literature review, a complete list of severities (degree of impact of the challenges) and the difficulties to resolve had to be created. These two variables (severity and difficulty) determine the priority of the challenge. As each challenge is unique, and we are only interested in getting to a value for a priority per challenge, there is no need for using a t-test. The unit of analysis is, therefore, a particular challenge and what is being analysed is the priority on a specific challenge. The data type used in phase 2, following the graph in Figure 3.4, would be continuous interval data.

The free statistical tool JASP (https://jasp-stats.org) and Microsoft Excel were selected as tools to analyse the findings. The JASP analytics tool was primarily utilised to validate the outputs from Microsoft Excel.

In phase 2 it is not the intention to infer conclusions from the findings, but purely to calculate the mean (μ) of the responses for difficulty and severity per challenge and then to calculate a priority factor for each of the challenges.

The reason for the use of mean (μ) is as a result of the expectation that the plotting of the data in a graph would be evenly spread among the mean. There is also an expectation of skewness, especially for very high or low Likert indicators and that outliers may be found. It was further expected that the data would mostly be symmetrical and therefore, the use of mean (McCluskey & Lalkhen, 2007). When the kurtosis (peakedness) is found to be very low or negative, it is an indication that the distribution became very flat and that respondents had varied opinions on the question.

Outliers were determined using the box and whiskers graph, utilising the formula: Q1 - 1.5(IQR) & Q3 + 1.5(IQR). Whenever outliers were found, they were ignored from the set, and the average calculation was taken on the rest of the result set. This insured that the set was kept symmetrical as far as possible.

The formula to determine the weighted priority was:

*Weighted priority of challenge = (((Severity amount + Difficulty amount)/ (Difficulty amount/Severity amount))/30)*10*

We multiply by ten at the end to get to a priority indicator out of 10. Using the above formula, the following table (Table 3.14) of priorities can be deduced, which would fit the balance between severity and difficulty.

Severity	Difficulty	Priority
5	1	10.00
4	1	6.67
5	2	5.83
5	3	4.44
4	2	4.00
3	1	4.00
5	4	3.75
5	5	3.33
4	3	3.11
4	4	2.67
3	2	2.50
4	5	2.40
3	3	2.00
2	1	2.00
3	4	1.75
3	5	1.60
2	2	1.33
2	3	1.11
2	4	1.00
2	5	0.93
1	1	0.67
1	2	0.50
1	3	0.44
1	4	0.42
1	5	0.40

Table 3.14: Priority assignment concerning severity and difficulty

The priority assigned to every challenge will assist in providing a useful prioritised list of solutions based on the challenges' priority.

3.7.6 **Phase 3 (AAC acceptability) Data analysis**

When an Agile coach has logged into the AAC and has gone through AAC and received the results at the end, they will be prompted to provide feedback on usability, value,

perceived accuracy and usefulness. As in phase 2, the data type will also be continuous interval data.

Four questions are described in Table 3.12 that were asked to approximately 30 Agile coaches and Scrum Masters to deduce whether the AAC is accurate, usable, user-friendly and valuable to the Agile coaching community. Each of the four questions tested a specific trade of the AAC and had a different weighting. Accuracy should be the first and highest weighted item as, without it, no one would think of using it. Secondly, usefulness: if it is not useful or beneficial to an Agile coach, they would not recommend it (valuable). Value is the third highest weighted trade. Fourthly, usability: it deals with the interaction of the user with the interface of the tool. If the experience is not intuitive and friendly, even though it is accurate and useful, it might affect acceptance. All of these trades together describe whether Agile coaches would find the AAC acceptable or not.

The unit of analysis is an instance of the AAC that was completed and had to be evaluated by an Agile specialist. This then led to the questions to be completed in Table 3.15.

Question	Trade	Weight
Did you find the AAC user interface intuitive and easy to navigate?	Usability	0.16
Would you recommend the AAC free assessment tool to other Agile coaches?	Valuable	0.21
Were the results of the assessment tool in line with what you expected (accuracy) it to be?	Accuracy	0.35
Were the results of the AAC assessment tool useful to you?	Usefulness	0.28

Table 3.15: Weightings assigned to AAC testing trades

As outlined in Phase 2 (Challenges Prioritisation) Data analysis above, the JASP statistical tool and Microsoft Excel will be used to analyse the data to ascertain whether the AAC is found to be acceptable or not.

The expectation from the data is that the four questions would be generally symmetrical and have a normal unimodal distribution for each of the questions, and this is what the data reflected. There were no extreme highs or lows in the data for the four questions. In general, a typical bell-shaped curve was realised, and outliers were purposefully not factored in. In respect to the use of mean, mode and median, the mean of the data was

used to calculate the acceptance of the AAC. There was no testing for outliers as if we found a specific survey response to be valid; all individual answers were accepted in the survey. This was done to enhance the accurate reflection of respondents on the AAC.

Standard deviation (σ) is a measure of spread, meaning how widely spread the data is for any specific set of answers. This spread reflected the reliability of the data. A general low standard deviation was expected (reflecting tightly clustered data around the mean), which is substantiated in chapter 6. The standard error of deviation reflects how accurate the mean is concerning samples from the larger Agile coaching population.

To regard the AAC as acceptable, the data needed to have a low standard deviation and the total acceptance value required to be larger than 3.5. Table 3.16 illustrates a perfect acceptance score. For the purpose of this study, a score more significant than 70% or 3.5 out of 5 would deem the AAC as acceptable among the Agile coaching community given that the analysis of the data has shown the data to be trustworthy.

Mean Result / Question	Weight	Weighted result
5	0.16	0.8
5	0.21	1.05
5	0.35	1.75
5	0.28	1.4
Acceptance Total		**5**

Table 3.16: Perfect acceptance score from the four questions

The AAC is hosted on the site www.agilesense.co.za\aac and the various questionnaires were also conducted there.

3.8 Validity and Ethical Considerations

Validity in research is crucial. The appropriateness of the measurements used (are the measures appropriate for the intended purpose?), the accuracy of the data analysis of the results and generalisations of the findings would otherwise all be in question, and this would, in turn, invalidate the research.

Saunders, Lewis and Thornhill (2016:202) differentiate between aspects of validity:

1. Measurement validity (face validity, construct validity, content validity and predictive validity)
2. Internal validity
3. External validity

Each of the validity aspects above is discussed below as it applies to this study.

3.8.1 **Internal Validity**

Saunders, Lewis and Thornhill (2016:718) define internal validity as “the extent to which findings can be attributed to interventions rather than any flaws in [the] research design.” Kitchenham et al. (2002:21) describe internal validity as the extent to which the research design and analysis may be compromised by the existence of unexpected sources of bias. Fraenkel and Wallen (2013:1694) define internal validity as “the degree to which observed differences in the dependent variable are directly related to the independent variable, not some other (uncontrolled) variable”.

Leedy and Omrod (2013:101) assert that internal validity speaks to both cause and effect relationships in the data, as well as general conclusions drawn from the data. The latter applies to all research. The former occurs, in general, more in exploratory research, but the fact that phases two (Challenges Prioritisation) and three (AAC acceptability) follow a non-probability volunteer approach could add to internal validity issues. The conclusions drawn from the data refers to the inferences a researcher could make every time an event could not be observed directly.

The study relies on the skill and experience of Scrum Masters and Agile coaches to add to the integrity of the results, and the study infers that such experience did happen. The online questionnaire requested that only Scrum Masters and Agile coaches with more than three years of experience complete the relevant surveys in order to increase internal validity.

Neither the respondents nor the author in this study had any knowledge of one another, and the respondents had no incentive to skew the results in any way. In general, if a respondent would want a tool like the AAC, they would also want it to be useful. They would subsequently not wish to create false inputs to the questionnaires.

Concerning questionnaires, Saunders, Lewis and Thornhill (2016:450) postulate that the ability of a questionnaire to measure its intended measurement is sometimes referred to as measurement validity. For validity concerns in questionnaires, researchers generally speak to content validity, criterion-related validity and construct validity (Saunders, Lewis & Thornhill, 2016:450).

3.8.2 **Measurement Validity**

Measurement validity assesses whether the intentions of the research are valid (Saunders, Lewis & Thornhill, 2016:202). In respect to phase one, the secondary data represented by the literature adequately covered the research question and objective.

3.8.2.1 Content Validity

Content validity in phases two and three refers to the extent that the questions in the questionnaires provided adequate coverage of the investigative questions. In phase 2, the question requiring an answer, is: “What is the general priority of the various Agile adoption challenges in software development industries?” The questionnaire directly deals with the two primary variables (difficulty and severity) for every challenge and therefore, fully adheres to content validity.

Phase three endeavour to answer the question: “Will an automated agile adoption framework be accurate, acceptable and useful to Agile coaches, helping companies adopting Agile software development?” The four questions posed in the questionnaire for phase three speak to covering these attributes. That includes usability, value, accuracy and usefulness also being indicators for acceptability.

3.8.2.2 Criterion-related Validity

Criterion-related validity speaks to “the ability of a statistical test to make accurate predictions from the data” (Saunders, Lewis & Thornhill, 2016:714). Phases 1 and 2 do not try to make predictions from the data, but phase 3, however, hypothesises that the AAC might be acceptable for the Agile coach community. Phase three data analysis (3.7.6) refers to the statistical tests in trying to corroborate that hypothesis.

3.8.2.3 Construct Validity

Construct validity refers to the extent to which a set of questions in a questionnaire measures the presence of the construct the researcher intends to measure (Oates, 2006; Saunders, Lewis & Thornhill, 2016:450). Construct validity attempts to ascertain how well one can generalise from a set of questions to the resulting construct. A short pilot survey questionnaire was done for both phases 2 and 3, incorporating a subset of the target population to increase the construct validity (Laanti et al., 2011:286). The pilot questionnaire was distributed to 10 respondents, and the feedback was incorporated into the applicable questionnaires and database before the full online survey for phases 2 and 3 was started.

3.8.3 **External Validity**

External validity is concerned with the question of whether a study's research results can be generalised to other sample groups for the same population. For example, if someone else would want to reproduce this study, would he or she have similar findings for phases 1, 2 and 3?

The amount of formal and informal literature concerning Agile adoption and how it relates to challenges and solutions would increase steadily as interest in Agile adoption increases. This factor would, however, not be a significant factor in deciding the external validity of Phase 1. Phase 1's sample selection includes several studies (see bottom of Phase 1 (Challenges & Solutions) Data collection, where the researchers conducted literature reviews on challenges and solutions. It is the author's view that with a possible different sample set very similar results would be achieved for phase 1 (Challenges and solutions). Even though the sampling design techniques followed in phases 2 and 3 follow a non-probability volunteer approach, the samples included global representatives for both Agile coaches and Scrum Masters in the study. As the AAC will be a living tool, the international community would be open to suggest a further enhancement to the database and user experience.

The same wider audience of people who have been approached for phase 2 has also been contacted in phase 3. About 80% of the people who have actually responded to phase 3 have also participated in Phase 2. This correlation, however beneficial, is consequential. The external validity should not be skewed if someone else executes

these surveys with different sample groups in phases 2 and 3 with a primary focus on the South African context. The differentiating factor or main uncertainty is the country. The results from coaches in the same country should normalise out with large enough distributions.

3.8.4 **Ethical Considerations**

This study involves three phases (Table 1.2), two of which require quantitative feedback from people. There are therefore specific ethical considerations which need to be considered, carefully evaluated and addressed. This study complied with the UNISA Research Ethics Policy (2007) and obtained ethics clearance, see appendix 8.2.

The literature review revealed that a quantitative study is necessary for the prioritisation of challenges in the literature (Phase 2 (Challenges Prioritisation)). During phase 2, respondents are requested to indicate a priority and difficulty rating for each of the 97 challenges identified. In phase 3, respondents need to evaluate the AAC to ascertain whether it would be acceptable for the Agile coaching community.

The procedure used in both phases 2 and 3 to collect data for the study were via online questionnaires. Various forums and mailing lists currently exist for a variety of Agile communities internationally. Table 3.7 provides an overview of the various areas that were canvassed for respondents. The author requested permission to post on these mailing lists or forums from the mailing list and forum owners. South African Agile forums were used to submit requests for participation in the study. On the online forum or mailing list, it was stipulated that the survey will respect the respondent's privacy and that the user may remain anonymous (which includes not sharing their name and surname) if they wanted to. No personal or organisational names are revealed as part of this study. The respondents were also informed that the data collected would not be used for anything other than this study, and they would have the right to withdraw at any stage of the research.

When the respondents to the survey (phase 2 or 3) clicked on the link provided, they were first confronted with a research information page and then an online consent form. The consent form needed to be accepted before they could continue to the survey. The

participant needed to acknowledge that he or she was giving consent to use his or her data in the study. It was stipulated that all data would be safely stored, not distributed to any third parties and would only be used for this study. If the user refused, he or she would be thanked for their willingness in considering to participate in the survey, and they were not able to gain access to the survey.

In case of phase 3, the respondent first had to complete the AAC process up to the end, at which point the user would be requested to provide quick feedback on the experience in respect to the questions presented in Table 3.12. The AAC asked questions regarding usefulness, accuracy and acceptability of the tool and data. This information was saved in a database, queried and analysed to determine the acceptability of the tool.

All respondents, irrespective of the phase they participated in, were thanked for their participation at the end of the survey. They were also informed that they would receive the analysed results at the completion of the study.

3.9 Summary

This research design chapter catered for the research design for three quantitative surveys:

1. Phase 1: Quantitative survey on the literature concerning Agile adoption challenges and possible solutions.

2. Phase 2: Quantitative survey using an online questionnaire to determine the priority on Adoption challenges. This priority is determined by the values of the dependent variables, severity and the difficulty of a challenge. The severity and difficulty are the main outputs of the actual survey.

3. Phase 3: Quantitative survey using an online questionnaire to determine the acceptability of the AAC.

Throughout this chapter, the research onion described by Saunders, Lewis and Thornhill (2016:124) was used to elaborate on the various skins of the “onion” as well as to explain why specific decisions on instruments and other research design criteria were made. The various selections in the research design are articulated in Table 1.3 for further reference.

Finally, various ethical and validity concerns are addressed. Special consideration is given to internal reliability via the Cronbach calculation in chapters four and five.

In data analysis chapters four and five, the research design elements come into fruition and analysis is done on the Agile challenges priority survey and acceptance on the AAC survey.

Chapter 4: AGILE CHALLENGES PRIORITY ANALYSIS AND COMPLETING AAC DATABASE

4.1 Introduction

In order to understand the priority of challenges which need to be displayed in the AAC, it is essential to analyse the results from the quantitative study on the severity and difficulty of Agile adoption challenges carefully. The analysis and results of this quantitative study answers the research sub-question:

b. What is the general priority of the various Agile adoption challenges in software development industries?

The word “general” in the research question is significant. The views of the agile coaches and scrum masters are quite subjective, and the aim is not to reach an absolute truth concerning severity or difficulty, but rather to determine a tendency towards the severity and difficulty of various challenges. It is known that Agile and companies using Agile evolve. New and better ways of adoption are found, and the current challenges may not be that relevant in the near future. Hoogendoorn (2012:2) makes the statement “Agile is an evolution rather than a revolution”, which describes the evolving nature of Agile.

Concerning this analysis, a positivist, inductive approach is applied to data from an online questionnaire. A non-probability sampling approach is used using the snowball and self-selection techniques.

In the end, 51 valid responses were received to the questionnaire from various Agile coaches and Scrum masters. The data analysis follows a continuous interval, descriptive analysis strategy. Section 8.5.1 showcases the questionnaire used for this survey utilizing the Likert scale to get answers for the severity and difficulty of specific challenges.

The analysis and the interpretation of the findings are broken down in four parts. In the first part, there is a general discussion on the analysis covering the pilot and response rate, reliability and internal consistency as well as the use of mean vs median. In the second part, the results of the analysis are presented concerning quantitatively analysing and interpreting the Agile adoption severity and difficulty results. In the third part, the focus is on specific statistical measurements including, skewness, kurtosis and standard

deviation. Lastly, the fourth part, a priority calculation, is done based on severity & difficulty outcomes. A summary of all the significant findings is finally presented in this chapter.

4.2 Pilot and Response Rate

A pilot of 10 volunteers was executed on the survey to enhance the construct validity. The pilot group faced various challenges:

1. They perceived the questionnaire mentally taxing as they first have to think what the challenge means for them, what the severity of impact that challenge could have on their company and then finally how difficult it would be to mitigate that challenge. This needed to be repeated 97 times.

2. There were some user experience issues which was fixed to assist them in getting through the survey quicker.

3. There were a few coding issues, concerning saving the feedback of the survey, which was rectified.

The pilot is an essential step to rectify various issues and to understand the respondent's stance towards the survey.

The response rate utilising self-selection and snowball techniques were initially positive, however not enough to come to reach a sufficient sample size. A target of 30 respondents was initially set. Unfortunately, due to the survey's mentally taxing nature, 35% of respondents did not complete the survey. There were also five surveys where the survey was completed unsatisfactorily. For example selections of a series of 1's and then 5's is clearly invalid. Those surveys are subsequently discounted. Multiple global mailing lists and forums were canvassed initially to join the survey, but minimal responses were received due to the taxing nature of the survey and secondly because they did not know the author. Subsequently, the broader South African Agile community was solicited employing WhatsApp, LinkedIn and email bore the most desired results, as the author was more known in the South African context.

There was very little response from the international community. There were only three responses from the Agile Spain Yahoo group. This sparse international input could skew the perception of the final prioritization in that it really does not present a global view but rather a picture on the priority of challenges in the South African context. We can therefore not generalise with respect to the priority on the worldwide community but more towards the South African community. The online survey ultimately yielded 51 valid responses among Scrum masters and Agile coaches. This constitutes a functional sample group, especially in a relatively small Agile community in South Africa.

4.3 Reliability and Internal Consistency

In the research methodology chapter, various reliability methods and concerns are discussed. In this chapter, the reliability area of internal consistency is further discussed concerning the calculating method Cronbach's Alpha (α). Saunders, Lewis and Thornhill (2016:451) stipulate that this method is the most common method used to estimate the internal consistency reliability (measuring the consistency of responses to a set of questions). They further establish that on a scale of 0 to 1, anything above 0.7 would indicate that the same thing is measured.

Chiu and Liu (20082016) break this down further to establish clear demarcated areas for what the results could indicate concerning reliability.

Cronbach α value	Interpretation
< 0.5	unacceptable
> 0.5	poor
> 0.6	questionable
> 0.7	acceptable
> 0.8	good
> 0.9	excellent

Table 4.1: Cronbach's α values interpretation

Cronbach α calculations have been done on the responses for both severity and difficulty on Agile adoption challenges:

Items (questions)	97

Sum of item variance	70.352941
The variance of the total score	817.41253
Cronbach's α	**0.9234523**

Table 4.2: Cronbach's α calculation for Severity instrument

Items (questions)	97
Sum of item variance	72.01076509
The variance of the total score	654.7550942
Cronbach's α	**0.899289807**

Table 4.3: Cronbach's α calculation for Difficulty instrument

Taking the groupings in Table 4.1 and Cronbach's values in Table 4.2 and Table 4.3 into account, it is clear that both values indicate a very high measure of internal consistency reliability.

These Cronbach's Alpha measurements were taken before the outlier items were factored in.

4.4 Use of mean vs median

Anderson et al. (2014) suggest that the median would be a better alternative to determine the central tendency in cases where the dispersion of data is positive or negatively skewed. As outliers are excluded from the mean taken in the column "(μ) excl, outliers" in Table 4.4 and Table 4.5, and the fact that the median and mean are generally very close to one another for both the severity and difficulty in remediating Agile adoption challenges, the author has selected not to use the median but rather the mean (μ) in both cases.

4.5 Quantitative interpretation of Agile adoption Severity results

In Table 4.4 below, the raw outputs of the analysis of the various challenges are indicated concerning the severity responses. The first step is to determine whether there are any outliers that would influence a symmetrical distribution of the severity outcome histogram considerably. The formula to determine the outlier for every challenge is Q1 - 1.5(IQR) & Q3 + 1.5(IQR).

Having obtained the outliers, the mean (μ) is calculated for every challenge (not factoring in the outliers). The results in the column "(μ) excl. outliers" displays the outcome of the calculation:

	Outlier Areas					Analysis						
ID	Q1	Q3	IQR	UB	LB	(μ) excl. outliers	Kurtosis	Median	(μ)	(σ)	σ excl. outliers	Skew-ness
4	3	3	0	3	3	3.00	1.21	3	2.73	0.78	0.00	-1.30
5	2	3	1	4.5	0.5	2.46	-0.32	3	2.86	0.78	0.97	0.50
6	2	3	1	4.5	0.5	2.82	-1.09	3	2.82	0.74	0.74	0.29
7	3	4	1	5.5	1.5	3.76	0.15	4	3.76	0.65	0.65	-0.17
8	4	4	0	4	4	4.00	0.33	4	3.96	0.56	0.00	-0.01
9	3	4	1	5.5	1.5	3.40	-0.34	3	3.12	1.11	0.84	-0.41
10	4	5	1	6.5	2.5	4.73	1.81	5	4.41	0.98	0.45	-1.66
11	3	5	2	8	0	4.35	-0.93	5	4.35	0.96	0.96	-0.89
12	2	3	1	4.5	0.5	2.61	-0.27	3	2.61	0.85	0.85	-0.54
13	4	4	0	4	4	4.00	1.38	4	3.86	0.94	0.00	-1.19
14	4	5	1	6.5	2.5	4.71	2.16	5	4.37	1.06	0.51	-1.71
15	4	4	0	4	4	4.00	3.05	4	3.88	1.07	0.00	-1.74
16	3	4	1	5.5	1.5	3.30	0.72	3	3.25	0.74	0.68	-0.44
17	3	4	1	5.5	1.5	3.67	0.27	4	3.67	0.55	0.55	-0.71
18	2	4	2	7	-1	3.06	-0.62	3	3.06	1.14	1.14	-0.44
19	3	3	0	3	3	3.00	-0.22	3	3.06	0.61	0.00	-0.03
20	4	5	1	6.5	2.5	4.35	2.56	4	4.24	0.89	0.69	-1.35
21	3	4	1	5.5	1.5	3.86	-0.67	4	3.86	0.66	0.66	0.15
22	3	5	2	8	0	3.84	-0.78	4	3.84	0.99	0.99	-0.44
23	2	3	1	4.5	0.5	2.25	-1.04	2	2.25	0.74	0.74	-0.44
24	2	3	1	4.5	0.5	2.37	-0.02	3	2.63	1.15	0.88	0.20
25	4	4	0	4	4	4.00	0.47	4	4.14	0.53	0.00	0.15
26	2	2	0	2	2	2.00	3.14	2	2.10	0.76	0.00	0.96
27	4	5	1	6.5	2.5	4.41	-0.53	5	4.41	0.67	0.67	-0.69
28	3	4	1	5.5	1.5	3.57	-0.75	4	3.57	0.92	0.92	-0.13
29	3	4	1	5.5	1.5	3.16	-0.71	3	3.12	0.82	0.77	-0.44
30	3.5	5	1.5	7.3	1.3	4.08	-0.81	4	4.08	0.82	0.82	-0.36
31	3	5	2	8	0	3.59	-0.80	4	3.59	1.24	1.24	-0.46
32	4	4	0	4	4	4.00	0.32	4	4.08	0.56	0.00	0.03
33	4	5	1	6.5	2.5	4.40	2.38	4	4.35	0.63	0.53	-0.91
34	1	2	1	3.5	-1	1.90	-1.20	2	1.90	0.76	0.76	0.16
35	2	4	2	7	-1	2.63	-0.76	2	2.63	0.94	0.94	0.95
36	3	4	1	5.5	1.5	3.06	-0.92	3	3.06	0.70	0.70	-0.08
37	1	2	1	3.5	-1	1.71	-0.69	2	1.71	0.50	0.50	-0.41
38	3	3	0	3	3	3.00	0.45	3	2.92	1.00	0.00	-0.58
39	3.5	4	0.5	4.8	2.8	3.97	0.16	4	3.39	1.15	0.16	-1.29

40	2	3	1	4.5	0.5	2.25	-0.58	2	2.25	0.80	0.80	0.00
41	3	4	1	5.5	1.5	3.63	0.27	4	3.63	0.77	0.77	-0.83
42	4	4	0	4	4	4.00	3.39	4	3.98	0.62	0.00	-1.03
43	2	2	0	2	2	2.00	0.13	2	1.94	0.58	0.00	0.00
44	2	3	1	4.5	0.5	2.46	-0.61	3	2.51	1.03	0.97	-0.14
45	3	4	1	5.5	1.5	3.38	-0.48	3	2.96	1.09	0.66	-0.66
46	4	5	1	6.5	2.5	4.47	-1.14	4	4.47	0.54	0.54	-0.26
47	4	4	0	4	4	4.00	0.61	4	4.16	0.50	0.00	0.28
48	1	2	1	3.5	-1	1.73	-0.72	2	1.73	0.67	0.67	0.36
49	1.5	3	1.5	5.3	-1	2.12	-0.52	2	2.12	0.86	0.86	0.34
50	3	4	1	5.5	1.5	3.69	-0.84	3	3.06	1.30	0.69	-0.49
51	3	4	1	5.5	1.5	3.69	0.49	4	3.69	0.76	0.76	-0.79
52	2	3	1	4.5	0.5	2.61	-0.86	3	2.61	0.98	0.98	-0.30
53	3	4	1	5.5	1.5	3.18	-0.43	3	3.18	0.62	0.62	-0.13
54	3	4	1	5.5	1.5	3.35	-0.42	3	3.35	0.80	0.80	0.01
55	4	5	1	6.5	2.5	4.25	-0.93	4	4.25	0.72	0.72	-0.41
56	2	2.5	0.5	3.3	1.3	2.29	0.17	2	2.12	0.68	0.46	0.23
57	3	5	2	8	0	3.92	0.41	4	3.92	1.28	1.28	-1.13
58	4	5	1	6.5	2.5	4.24	3.07	4	4.18	0.82	0.69	-1.22
59	1	2.5	1.5	4.8	-1	1.84	-0.51	2	1.84	0.95	0.95	0.75
60	2	3	1	4.5	0.5	2.26	-0.16	2	2.31	0.95	0.88	0.19
61	3	4	1	5.5	1.5	3.56	-0.17	3	3.16	1.12	0.67	-0.83
62	3	5	2	8	0	3.98	0.02	4	3.98	1.07	1.07	-0.86
63	1	3	2	6	-2	2.04	-1.30	2	2.04	0.77	0.77	-0.07
64	1	3	2	6	-2	2.24	-0.55	3	2.24	0.97	0.97	0.04
65	2	3	1	4.5	0.5	2.22	0.26	2	2.27	0.90	0.82	0.28
66	1	2	1	3.5	-1	1.63	-0.63	2	1.63	0.60	0.60	0.35
67	3	4	1	5.5	1.5	3.52	0.37	3	3.18	1.03	0.59	-1.02
68	2	4	2	7	-1	3.22	-0.89	3	3.22	0.94	0.94	-0.30
69	2.5	3	0.5	3.8	1.8	2.73	2.81	3	2.82	0.59	0.45	0.64
70	3	4	1	5.5	1.5	3.59	0.47	4	3.59	0.61	0.61	-1.16
71	3	4	1	5.5	1.5	3.69	-0.03	4	3.69	0.86	0.86	-0.69
72	3	4	1	5.5	1.5	3.86	-1.01	4	3.86	0.72	0.72	0.21
73	3	4	1	5.5	1.5	3.76	-0.35	4	3.76	0.59	0.59	0.08
74	4	4	0	4	4	4.00	2.41	4	4.04	0.45	0.00	0.19
75	2	5	3	9.5	-3	3.98	-0.94	5	3.98	1.39	1.39	-0.86
76	3	3	0	3	3	3.00	3.11	3	2.84	0.58	0.00	0.63
77	1	2	1	3.5	-1	1.72	0.02	2	1.76	0.76	0.70	0.69
78	4	5	1	6.5	2.5	4.44	0.85	4	3.92	1.32	0.55	-1.37
79	2	3	1	4.5	0.5	2.29	-1.18	2	2.29	0.46	0.46	0.90
80	4	5	1	6.5	2.5	4.30	9.14	4	4.24	0.68	0.51	-1.86
81	3	4	1	5.5	1.5	3.41	-0.78	3	3.41	0.57	0.57	-0.30
82	1	2	1	3.5	-1	1.64	0.06	2	1.69	0.79	0.72	0.86
83	1	2	1	3.5	-1	1.92	1.83	2	1.98	0.84	0.72	0.86
84	3	4	1	5.5	1.5	3.57	-0.23	4	3.57	0.81	0.81	-0.46
85	3	4	1	5.5	1.5	3.14	0.10	3	3.10	0.73	0.67	-0.46
86	2	3	1	4.5	0.5	2.20	-0.70	2	2.20	0.66	0.66	-0.23
87	2	4	2	7	-1	3.08	-0.91	3	3.08	1.23	1.23	-0.08
88	2	3	1	4.5	0.5	2.67	-0.60	3	2.67	0.59	0.59	0.22

89	3	4	1	5.5	1.5	3.47	-0.32	4	3.47	0.64	0.64	-0.80
90	3	4	1	5.5	1.5	3.46	2.18	3	3.41	0.67	0.58	-0.69
91	2	4	2	7	-1	2.78	-0.87	3	2.78	1.29	1.29	0.29
92	3	3	0	3	3	3.00	-0.33	3	2.75	1.09	0.00	-0.41
93	3	4	1	5.5	1.5	3.69	0.81	4	3.69	0.73	0.73	-0.97
94	3	4	1	5.5	1.5	3.57	-0.14	4	3.57	0.81	0.81	-0.69
95	4	5	1	6.5	2.5	4.55	0.05	5	4.20	1.06	0.63	-1.11
96	1	2	1	3.5	-1	1.78	2.00	2	1.82	0.62	0.55	0.63
97	2	2	0	2	2	2.00	12.69	2	2.12	0.38	0.00	3.39
98	2	3	1	4.5	0.5	2.08	2.40	2	2.14	0.78	0.67	0.80
99	2	3.5	1.5	5.8	-0	2.73	-0.39	2	2.73	0.98	0.98	0.96
100	1	3	2	6	-2	2.04	-0.26	2	2.04	1.00	1.00	0.53

Table 4.4: Severity analysis of Agile challenges

4.6 Quantitative interpretation of Agile adoption Difficulty results

The same process of analysis is followed as with the severities' analysis above. That is, the first focus is on determining the outliers in order to align closest to a symmetrical Bell curve. The five columns under "Outlier Areas" in Table 4.5 is used to calculate the outliers. Excluding the outliers, the mean difficulty is calculated for the various Agile challenges. These values are represented in column "(μ) excl. outliers" of Table 4.5 below:

	Outlier Areas					Analysis						
ID	Q1	Q3	IQR	UB	LB	(μ) excl. outliers	Kurtosis	Median	(μ)	(σ)	σ excl. outliers	Skew-ness
4	1	3	2	6	-2	2.04	-0.13	2	2.04	1.09	1.09	0.87
5	2	2.5	0.5	3.25	1.25	2.75	-0.13	2	2.08	0.72	0.44	0.22
6	2	3	1	4.5	0.5	2.77	0.68	3	2.94	0.88	0.67	0.67
7	1.5	2	0.5	2.75	0.75	1.7	4.04	2	1.94	0.76	0.46	1.24
8	3	4	1	5.5	1.5	3.53	-0.38	4	3.53	0.81	0.81	-0.10
9	2	3	1	4.5	0.5	2.49	-0.61	2	2.49	0.88	0.88	0.31
10	3	5	2	8	0	3.92	-0.48	4	3.92	0.98	0.98	-0.64
11	4	5	1	6.5	2.5	4.47	-0.56	5	4.47	0.81	0.81	-1.08
12	2	3	1	4.5	0.5	2.57	-0.66	3	2.57	0.88	0.88	0.06
13	4	4	0	4	4	4	6.98	4	3.96	0.69	0	-1.84
14	3	4	1	5.5	1.5	3.89	0.58	4	3.55	1.15	0.71	-1.06
15	1	2	1	3.5	-0.5	1.56	0.06	1	1.61	0.83	0.76	1.08
16	2	3	1	4.5	0.5	2.35	-0.34	2	2.35	0.74	0.74	-0.08
17	2	4	2	7	-1	2.78	-0.83	3	2.78	1.25	1.25	0.24
18	2	3	1	4.5	0.5	2.33	0.15	2	2.33	0.59	0.59	0.39
19	3	3	0	3	3	3	2.81	3	3.1	0.73	0	1.46
20	3	4	1	5.5	1.5	3.66	-0.24	3	3.61	0.94	0.87	0.13

21	4	5	1	6.5	2.5	4.54	7.6	5	4.47	0.76	0.58	-2.19
22	2	3	1	4.5	0.5	2.39	-0.48	2	2.39	0.8	0.8	-0.12
23	2	2	0	2	2	2	-0.06	2	2.04	0.6	0	-0.01
24	1.5	2.5	1	4	0	1.89	1.47	2	2.2	1.13	0.67	1.32
25	3	4	1	5.5	1.5	3.63	-0.07	4	3.63	0.69	0.69	-0.10
26	1	2	1	3.5	-0.5	1.86	-0.45	2	1.86	0.63	0.63	0.11
27	3	5	2	8	0	3.8	-0.78	4	3.8	1	1	-0.46
28	4	5	1	6.5	2.5	4.22	0.86	4	4.18	0.68	0.62	-0.63
29	2	4	2	7	-1	3.04	-0.4	3	3.04	0.87	0.87	0.49
30	4	5	1	6.5	2.5	4.33	-0.66	4	4.33	0.65	0.65	-0.46
31	3	4	1	5.5	1.5	3.53	-0.89	3	3.43	1.15	1.06	-0.19
32	4	5	1	6.5	2.5	4.39	0.37	4	4.16	0.95	0.65	-1.06
33	4	5	1	6.5	2.5	4.43	-0.75	4	4.43	0.57	0.57	-0.38
34	2	2	0	2	2	2	13.18	2	2.08	0.39	0	2.82
35	2	4	2	7	-1	2.8	-0.94	2	2.8	0.96	0.96	0.69
36	1	2	1	3.5	-0.5	1.84	-0.74	2	1.84	0.67	0.67	0.20
37	1	2	1	3.5	-0.5	1.38	4.48	1	1.43	0.61	0.49	1.66
38	3	4	1	5.5	1.5	3.3	0.31	3	2.98	0.95	0.55	-0.98
39	3	4	1	5.5	1.5	3.36	0.03	3	3.04	1.02	0.65	-0.79
40	2	2	0	2	2	2	1.9	2	1.88	0.62	0	0.60
41	4	4.5	0.5	5.25	3.25	4.33	-0.83	4	4.04	0.69	0.47	-0.05
42	2	3	1	4.5	0.5	2.59	-0.53	3	2.59	0.64	0.64	0.62
43	2	4	2	7	-1	2.98	-0.69	3	2.98	0.79	0.79	-0.22
44	2	3	1	4.5	0.5	2.53	0.03	3	2.53	0.73	0.73	-1.23
45	2	3	1	4.5	0.5	2.78	-1.24	3	2.78	0.78	0.78	0.41
46	3	3	0	3	3	3	4.91	3	2.8	0.53	0	0.65
47	3	4	1	5.5	1.5	3.27	-0.8	3	3.27	0.78	0.78	-0.26
48	1	2	1	3.5	-0.5	1.57	-0.56	2	1.57	0.61	0.61	0.56
49	2	3	1	4.5	0.5	2.12	-0.53	2	2.12	0.82	0.82	0.24
50	2	3	1	4.5	0.5	2.73	2.53	3	2.82	0.68	0.53	1.02
51	3	3.5	0.5	4.25	2.25	3.3	-0.13	3	3.08	0.72	0.46	0.22
52	2	3	1	4.5	0.5	2.45	-0.48	2	2.45	0.58	0.58	0.20
53	3	4	1	5.5	1.5	3.58	-0.26	4	3.53	0.97	0.91	-0.29
54	4	4	0	4	4	4	0.98	4	3.59	1.17	0	-1.39
55	3	4	1	5.5	1.5	3.69	-0.26	4	3.69	0.91	0.91	-0.67
56	2	3	1	4.5	0.5	2.2	-0.85	2	2.2	0.69	0.69	-0.28
57	2	3.5	1.5	5.75	-0.25	3.06	-0.35	3	3.06	1.1	1.1	0.35
58	3	5	2	8	0	4.04	-0.89	4	4.04	0.82	0.82	-0.30
59	1	2	1	3.5	-0.5	1.45	-0.85	1	1.45	0.54	0.54	0.60
60	3	4	1	5.5	1.5	3.22	-0.42	3	3.22	0.78	0.78	0.12
61	3	5	2	8	0	3.78	-1.35	4	3.78	1.05	1.05	-0.09

62	3	4	1	5.5	1.5	3.68	-0.57	4	3.63	1.04	0.98	-0.30
63	3	3	0	3	3	3	0.98	3	2.82	0.43	0	-0.95
64	3	3	0	3	3	3	0.29	3	2.98	1.09	0	-0.06
65	1	3	2	6	-2	2.51	-0.39	3	2.51	1.22	1.22	0.28
66	2	3	1	4.5	0.5	2.3	0.5	2	2.57	1.08	0.76	0.75
67	1	2	1	3.5	-0.5	1.82	-1.32	2	1.82	0.79	0.79	0.33
68	3	4	1	5.5	1.5	3.93	0.59	4	3.65	1.15	0.77	-0.92
69	3	5	2	8	0	3.53	-0.9	4	3.53	1.33	1.33	-0.49
70	2	3	1	4.5	0.5	2.84	-1.19	3	2.84	0.76	0.76	0.27
71	3	4	1	5.5	1.5	3.49	-0.58	4	3.49	0.58	0.58	-0.60
72	2	3	1	4.5	0.5	2.49	1.9	3	2.69	0.86	0.55	1.45
73	4	5	1	6.5	2.5	4.57	-0.26	5	4.22	1.1	0.7	-1.10
74	2	4.5	2.5	8.25	-1.75	3.18	-0.96	3	3.18	1.34	1.34	-0.02
75	2	5	3	9.5	-2.5	3.73	-1	4	3.73	1.22	1.22	-0.62
76	3	3	0	3	3	3	0.83	3	3.22	0.81	0	1.00
77	1	2	1	3.5	-0.5	1.56	6.98	2	1.63	0.75	0.58	1.94
78	2	3.5	1.5	5.75	-0.25	3.02	-0.28	3	3.02	0.81	0.81	0.43
79	3	4	1	5.5	1.5	3.2	0.13	3	3.2	0.78	0.78	0.44
80	3	4	1	5.5	1.5	3.78	0.75	4	3.78	0.67	0.67	-0.54
81	1	2	1	3.5	-0.5	1.56	-0.29	1	1.61	0.87	0.81	1.06
82	1	1	0	1	1	1	1.71	1	1.41	0.8	0	1.74
83	1	2	1	3.5	-0.5	1.4	1.29	1	1.45	0.81	0.73	1.59
84	2	4	2	7	-1	3.16	-0.9	4	3.16	1.3	1.3	-0.93
85	2	4	2	7	-1	2.53	-0.7	2	2.53	1.17	1.17	0.70
86	2	2.5	0.5	3.25	1.25	2.3	-0.34	2	2.12	0.62	0.46	-0.08
87	1	2	1	3.5	-0.5	1.78	1.15	2	2.14	1.18	0.67	1.31
88	1	3	2	6	-2	2.02	0.36	1	2.02	1.29	1.29	1.13
89	2	3	1	4.5	0.5	2.43	-0.73	2	2.43	0.54	0.54	0.68
90	3	4	1	5.5	1.5	3.52	-0.5	3	3.47	1.01	0.95	0.14
91	2	3	1	4.5	0.5	2.22	-0.59	2	2.22	0.64	0.64	-0.22
92	2	2	0	2	2	2	-0.21	2	1.9	0.61	0	0.05
93	3	4.5	1.5	6.75	0.75	3.67	-1.08	4	3.67	0.97	0.97	0.05
94	2	2	0	2	2	2	10.03	2	2.18	0.59	0	2.38
95	3	4	1	5.5	1.5	3.73	-0.46	4	3.73	0.78	0.78	0.00
96	1.5	2	0.5	2.75	0.75	1.72	5.44	2	1.9	0.76	0.46	1.62
97	1	2	1	3.5	-0.5	1.76	1.24	2	1.8	0.66	0.59	0.66
98	1	2	1	3.5	-0.5	1.67	2.09	2	2.06	1.19	0.52	1.66
99	2	3	1	4.5	0.5	2.22	0	2	2.22	0.67	0.67	0.13
100	1	2	1	3.5	-0.5	1.73	-0.42	2	1.73	0.57	0.57	0.04

Table 4.5: Difficulty analysis of Agile challenges

4.7 Skewness

The skewness values falling in the range -0.5 to 0.5 are generally seen as supporting a normal distribution (GoodData, 2018). A normal distribution is typically symmetrical on the mean point, and the mean and median are very close or the same. Another indicator of the normal distribution is how the data is distributed: one (68% of data points), two (98% of data points) and three (99% of data points) standard deviations from the mean. As can be seen in the highlighted values under the “Skewness” column, 50 challenges can be considered as normally distributed for the severity skewness measurement.

As the range of values is limited between 1 and 5, there is an expectation of skewness for both severity and difficulty values - especially for mean values close to 1 and 5.

Skewness is calculated without removing the outliers. If the outliers are to be removed, there would be more Agile challenges that follow the symmetrical Bell curve. This can also be seen comparing the columns “(μ) excl. outliers” and “Median” with one another. These values tend to be very close to one another. Sixteen of them are precisely the same.

In the case of “difficulty”, there are 48 highlighted values under the “Skewness” column. These values can be considered as normally distributed.

As the range of values is limited between 1 and 5, there was an expectation of skewness, especially for severity mean values close to 1 and 5.

4.8 Kurtosis

The Kurtosis of distribution is classified into three groups (Statistics How To, 2018):

1. Mesokurtic: excess kurtosis of zero or close to zero.
2. Platykurtic: negative excess kurtosis.
3. Leptokurtic: positive excess kurtosis.

Interestingly for severity, seven of the 97 challenges have kurtosis levels of less than -1, which falls in the platykurtic range. Concerning the seven, it typically alludes to a very flat

distribution over a series of values (two to five). This indicates that among the related seven respondents, there is no clear consensus on the actual severity of the particular challenges. It could also mean, given their companies' specific circumstances, that specific severity is seen as the higher or lower impact for them than in other companies.

In all of these cases, the normal mean and the mean without the outliers are precisely the same. This indicates that no outliers are identified, which underscores the even distribution.

For the remaining 90 challenges in severity, the kurtosis values indicate that the distribution curve is in general quite pointy, with an average kurtosis value of 0.36.

In comparison with the severities, for "difficulty" there was more or less the same (only six) number of challenges found with Kurtosis values of less than -1. This indicates the respondents were generally in consensus with the allocations given to the challenges, except for the six mentioned.

The average of the Kurtosis values come to 0.56. This is just above the value of a typical normal distribution (Mesokurtic). Twelve challenges have Kurtosis values of more than two, which represents significant "peakedness" for those symmetrical curves (Leptokurtic distribution).

4.9 Standard deviation

Standard deviation (σ) measures how dispersed the set of values are from the mean (Dodge, 2003). The smaller the number of σ, the closer the data points are to the mean.

Only 23 of the 97 challenges in severity have standard deviations of more than one. This means that only 23 challenges have cases where *not* 68% of the data points are covered in one point distribution off the mean. The highest value of σ is merely 1.39. Excluding the outliers in the standard deviation equation, merely eight values that are higher than 1 is present. These small standard deviation values indicate that the standard deviations of the severity values found for the various challenges are very close to the mean.

The "ID" column can be used to reference the actual challenges in Table 6.3.

The standard deviation characteristics of the difficulty measurement are very similar to those of the standard deviations of the severity of challenges. Twenty-four challenges have a standard deviation of more than one, with the highest value being 1.34. Excluding the outliers, the number decreases, and merely 13 challenges with a value larger than one are left. The most significant number for standard deviation in this group is also 1.34.

From this, it can safely be assumed that the values for difficulty are tightly dispersed from the mean and an assumption can be made that the data from the respondents concerning the difficulty in dealing with Agile adoption challenges is statistically consistent.

4.10 Priority calculation based on Severity & Difficulty

Having a reasonably reliable result set from the survey on severities and difficulties, the priority of challenges can be calculated. This priority will be used to display the challenges whenever the AAC is used. The priority of Agile adoption challenges is one of the independent variables metadata factors referenced in Figure 2.4, talking to the conceptual framework of the study.

Section 3.7.5 elaborates on the formula used for the calculation as well as possible distributions of priority indicated in Table 3.14. These distributions have come to be quite difficult to achieve in particular combinations, especially in generating high priority values. Looking at Table 4.6 below, it is clear that there is generally a definite correlation between the severity and the difficulty. That is, the higher the severity, the higher the general difficulty and vice versa.

Severity	Avg Difficulty
>4	3.55
>3	2.99
>2	2.43
<2	1.60

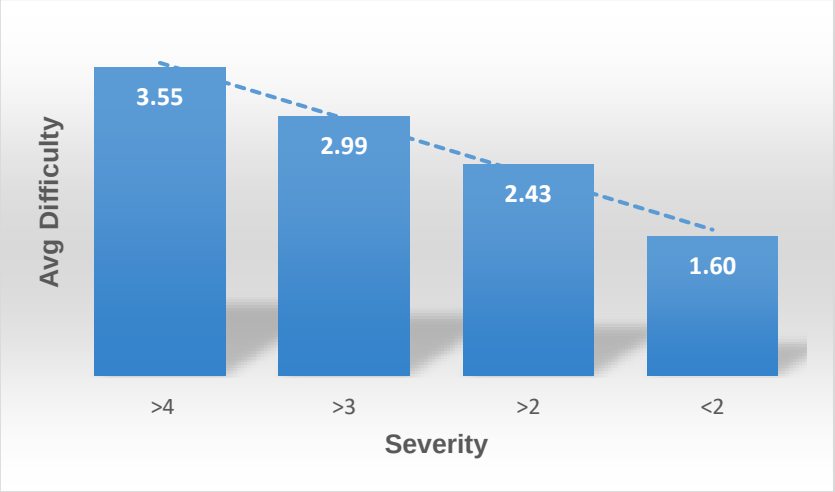


Table 4.6: Relationship between Severity and Average Difficulty

Similarly, the exercise is repeated for difficulty over average severity, and very similar results are achieved as can be seen in Table 4.7 below:

Difficulty	Avg Severity
>4	3.93
>3	3.63
>2	2.88
<2	2.43

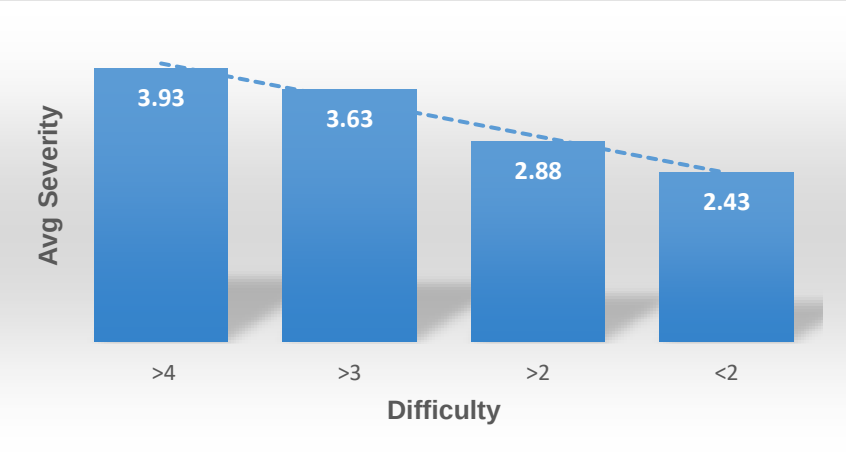


Table 4.7: Relationship between Difficulty and Average Severity

The tendencies in Table 4.6 and Table 4.7 help to explain the results of the priorities listed in Table 4.8. There are no priorities above five, as there are very high severities that have straightforward solutions to them. There are therefore no challenges which have a very low difficulty to remediate and a very severe impact.

After analysis, Table 4.8 provides the answer to the research question (1b) posed at the start of section 4.1. Table 4.8 took as input the severity μ values (excluding outliers) from Table 4 and the difficulty μ values (excluding outliers) from Table 5 to determine the priority.

It subsequently makes sense not to look at priority in isolation, but also to consider the factors of severity and difficulty as part of the AAC. The AAC does prioritisation but also shows the severity and difficulty in order to allow the user to decide whether he or she wants to change the priority subjectively for themselves, or use what is suggested.

Id	Severity	Difficulty	Priority
4	3.00	2.04	**2.47**
5	2.46	2.75	**1.55**
6	2.82	2.77	**1.90**
7	3.76	1.70	**4.03**
8	4.00	2.08	**3.90**
9	3.40	2.49	**2.68**
10	4.73	3.92	**3.48**
11	4.35	4.47	**2.86**
12	2.61	2.57	**1.75**
13	4.00	4.00	**2.67**
14	4.71	3.89	**3.47**
15	4.00	1.56	**4.75**
16	3.30	2.35	**2.64**
17	3.67	2.78	**2.84**
18	3.06	2.33	**2.36**
19	3.00	3.00	**2.00**
20	4.35	3.66	**3.17**
21	3.86	4.54	**2.38**
22	3.84	2.39	**3.33**
23	2.25	2.00	**1.59**
24	2.37	1.89	**1.78**
25	4.00	3.63	**2.80**
26	2.00	1.86	**1.38**
27	4.41	3.80	**3.17**
28	3.57	4.22	**2.20**
29	3.16	3.04	**2.15**
30	4.08	4.33	**2.64**
31	3.59	3.53	**2.41**
32	4.00	4.39	**2.55**
33	4.40	4.43	**2.92**
34	1.90	2.00	**1.24**
35	2.63	2.80	**1.70**
36	3.06	1.84	**2.71**

Id	Severity	Difficulty	Priority
52	2.61	2.45	**1.80**
53	3.18	3.58	**2.00**
54	3.35	4.00	**2.05**
55	4.25	3.69	**3.05**
56	2.29	2.20	**1.56**
57	3.92	3.06	**2.98**
58	4.24	4.04	**2.90**
59	1.84	1.45	**1.39**
60	2.26	3.22	**1.28**
61	3.56	3.78	**2.30**
62	3.98	3.68	**2.76**
63	2.04	3.00	**1.14**
64	2.24	3.00	**1.30**
65	2.22	2.51	**1.39**
66	1.63	2.30	**0.93**
67	3.52	1.82	**3.44**
68	3.22	3.93	**1.95**
69	2.73	3.53	**1.61**
70	3.59	2.84	**2.71**
71	3.69	3.49	**2.53**
72	3.86	2.49	**3.28**
73	3.76	4.57	**2.28**
74	4.00	3.18	**3.01**
75	3.98	3.73	**2.74**
76	3.00	3.00	**2.00**
77	1.72	1.56	**1.21**
78	4.44	3.02	**3.66**
79	2.29	3.20	**1.31**
80	4.30	3.78	**3.06**
81	3.41	1.56	**3.62**
82	1.64	1.00	**1.44**
83	1.92	1.40	**1.52**
84	3.57	3.16	**2.54**

37	1.71	1.38	**1.28**
38	3.00	3.30	**1.91**
39	3.97	3.36	**2.89**
40	2.25	2.00	**1.59**
41	3.63	4.33	**2.23**
42	4.00	2.59	**3.39**
43	2.00	2.98	**1.11**
44	2.46	2.53	**1.62**
45	3.38	2.78	**2.49**
46	4.47	3.00	**3.71**
47	4.00	3.27	**2.96**
48	1.73	1.57	**1.21**
49	2.12	2.12	**1.41**
50	3.69	2.73	**2.89**
51	3.69	3.30	**2.61**
85	3.14	2.53	**2.35**
86	2.20	2.30	**1.44**
87	3.08	1.78	**2.81**
88	2.67	2.02	**2.07**
89	3.47	2.43	**2.81**
90	3.46	3.52	**2.29**
91	2.78	2.22	**2.09**
92	3.00	2.00	**2.50**
93	3.69	3.67	**2.47**
94	3.57	2.00	**3.31**
95	4.55	3.73	**3.37**
96	1.78	1.72	**1.21**
97	2.00	1.76	**1.42**
98	2.08	1.67	**1.56**
99	2.73	2.22	**2.03**
100	2.04	1.73	**1.48**

Table 4.8: Priorities of challenges

Looking at the ten highest priority challenges in Table 4.9 below, it is clear why a number of them are on this list. The framework indicates a lack of training could have a severe impact on businesses, but to roll out training to mitigate this real risk is potentially not very difficult to do.

The second highest priority also makes sense. It could have a significant impact on the company if the developers do not understand the business needs, and therefore do not develop according to the real requirement. To mitigate this could generally be resolved by bringing business closer to the team: ensure the presence of an active and passionate product owner who could represent the business needs to the development team.

Each of these top 10 priority challenges can be explained similarly:

Id	Challenge	Severity	Difficulty	Priority
15	Lack of training.	4.00	1.56	4.75
7	Lack of business knowledge among developers.	3.76	1.70	4.03
8	Practices are prioritized over the Agile values and principles, and we do not understand it well enough.	4.00	2.08	3.90
46	Customer or client requirement to be part of the team.	4.47	3.00	3.71

78	Teams are split, and critical resources are removed after a successful pilot.	4.44	3.02	3.66
81	It can be difficult to convince management of the need for a new development approach.	3.41	1.56	3.62
10	The implications of devolved decision-making (Team making decisions).	4.73	3.92	3.48
14	Lack of management support.	4.71	3.89	3.47
67	The perception that predictability has decreased in Agile.	3.52	1.82	3.44
42	Short, iterative and incremental deliveries are a challenge.	4.00	2.59	3.39

Table 4.9: Highest 10 priority Challenges

4.11 Summary

The findings on the Agile challenges' priority was analysed in this chapter concerning the severity or impact of a specific Agile adoption challenge, the difficulty in resolving such a challenge and finally, a priority was deduced using a formula.

The linear relationship between severity and difficulty was looked at, and it was deduced that one could not look at priority in isolation, but still need to be cognizant of the severity and difficulty additionally before deciding on a specific priority.

Utilising the Cronbach's α method very high internal consistency reliability values were achieved for both severity and difficulty.

Analysing the Kurtosis for severity and difficulty, it was found that there are very few challenges with levels less than -1 (indicating flatness of the curve on the graph). In general, the average Kurtosis for severity is 0.36 and 0.57. This shows a general "peakedness" for both these independent variables and lays very close to the mesokurtic distribution.

Another exciting finding concerning the standard deviation (σ), was that for both severity and difficulty, the value of σ was meagre (never more than 1.39). The average of σ for severity was only 0.83 and for difficulty 0.84. This indicates the data points are concentrated very close to the mean and therefore suggests that the respondents in the sample group are very much consistent in their views of specific severities and difficulties about particular challenges.

It can finally be concluded that the research subquestion to determine the general priority of the various Agile adoption challenges in software development industries has been fulfilled.

Chapter 5: AAC USAGE DATA ANALYSIS AND INTERPRETATION OF RESULTS

5.1 Introduction

In this chapter, the main research question is addressed:

"Will an automated agile adoption framework be accurate, acceptable and useful to Agile coaches, helping companies adopting Agile software development?"

AAC is the end goal of this study. If it is to be used in an automated or manual fashion, it needs to be accurate, useful and acceptable in the Agile community.

The analysis in chapter 4 follows an inductive approach where a general theory of severity, difficulty and priority of challenges is formed. Similarly, through the literature review, a general theory in the framework (where a mapping between challenges and possible solutions were established) is devised. Chapter 5, however, takes as input these theories, and through the analysis in this chapter tries to deduce whether the research question postulated is valid or not.

The AAC is not meant to be a stagnant tool, but an evolving one - like the philosophy and framework it is representing. The Agile community inputs, based on usage, will direct this tool's life.

Concerning this analysis, a positivist and deductive approach on data from an online survey were followed. A non-probability sampling approach using the snowball and self-selection techniques was used. Section 8.5.2 showcases the questionnaire used for this survey after a respondent has completed an instance of the AAC.

The same Agile community from the priority survey was solicited to respond to the AAC acceptability survey. Fewer valid responses from various Agile coaches and Scrum masters were received than with the priority survey. A reason for this could be that the respondents thought that this survey would be just as onerous as the priority survey, and therefore elected not to participate. The data analysis also followed a continuous interval, descriptive analysis strategy.

The analysis and the interpretation of the findings are broken down in three phases. We will first be looking at feedback from the pilot as well as analysing the response rate, secondly analysing the reliability and internal consistency and thirdly interpreting the survey analysis with a specific focus on the mean, kurtosis and standard deviation. This chapter will finally be summarizing the key findings.

5.2 Pilot and Response Rate

As indicated in section “Measurement Validity” a pilot with ten volunteers was executed to test and ensure and enhance the construct validity. The pilot group faced only a few challenges, but mitigating those challenges assisted in improving the usability of the tool:

1. They found the tool not responsive enough concerning completing the survey on mobile devices. This issue was fixed quickly, and the relevant respondents acknowledged better user experience after the fix.

2. Some respondents complained that there were redundant questions in the user information page that were already covered in the survey agreement page. Those redundancies were also removed.

3. Some users found that explanations were not clear enough, and they, therefore, struggled to understand what needed to be done. Clearer descriptions were subsequently added to the various pages to ensure that there is no ambiguity and everyone is clear on what needs to be achieved.

The mitigation steps leading from the pilot enhanced the integrity and usability of the tool considerably and was an essential step in getting to a trustworthy survey on the acceptability of the AAC.

The response rate utilising self-selection and snowball techniques was a challenge from the start. As mentioned before, the same Agile community was solicited for the AAC survey as was used for the priority survey. Due to the significant mental focus required for the priority survey, there was not a quick uptake for the AAC survey by the Agile community. Several invitations were sent before a satisfactory response rate was attained. There were five surveys where the responses were all 1’s, 2’s or 3’s. There were

three surveys that had a minimal selection of challenges (one to three). These surveys were also discarded as they did not test the AAC in full.

Similar to the priority survey, the South African Agile community was the most significant contributors to the survey, and there was little response from the international community. There were only two respondents from Europe who completed the survey using links on Agile community forums.

There were ultimately 42 valid survey responses that were sufficient and used to test the acceptability of the AAC.

5.3 Reliability and internal consistency

Cronbach's Alpha (α) calculation method to estimate the internal consistency reliability is again utilised for the final survey. This method is to test whether there is some consistency in the responses from the Agile community.

Items (questions)	4
Sum of item variance	1.2545
The variance of the total score	2.6446
Cronbach's α	**0.7008**

Table 5.1: Cronbach's α for estimating internal consistency on AAC acceptability

The Cronbach's α value of 0.7008 in Table 5.1 above is seen as acceptable according to Table 4.1, discussing the meaning of the range of Cronbach's α values. Chow and Cao (2008:964) and Mitchell (1996) state that a value higher than 0.5 can be regarded as an acceptable internal consistency.

Taking into account the groupings in Table 4.1 as well as Cronbach's value in Table 5.1, it is clear that the value indicates an acceptable measure of internal consistency reliability.

This Cronbach's α measurement is taken without any outliers factored in.

5.4 Quantitative Interpretation of AAC survey results

In Table 5.2 below the raw outputs of the analysis on the acceptability of the AAC are indicated. As indicated before, in this instance, outliers are not factored in as an accurate

reflection of the respondents' feedback, and whether or not it was a valid submission, was sought. The anticipation for this descriptive analysis was that the results would be symmetrically distributed around the mean in a typical Bell curve. Besides the mean, the analysis functions kurtosis and the standard deviation were also selected to describe the results.

The results in the column "(μ)" display the final mean results of every question in the calculation:

Kurtosis	Median	(μ)	(σ)	Skewness
-0.72	4	4.31	0.52	0.28
-0.50	4	4.19	0.63	-0.17
-0.68	4	4.38	0.58	-0.28
-0.85	4	4.33	0.53	0.19

Table 5.2: AAC acceptability analysis on questions

Column "(μ)" is not, however, the final result of the findings. Taking into account the various questions' weightings, given in Table 3.16, a further calculation step is required.

Table 5.3 factors in the weighting for every question and then displays the weighted result for every question. Finally, the acceptability percentage achieved is indicated. An acceptability percentage of 86.32% has been obtained for the AAC from the 42 respondents that took part in the survey.

	Mean Result / Question	Weight	Weighted result
Q1	4.31	0.16	0.69
Q2	4.19	0.21	0.88
Q3	4.38	0.35	1.53
Q4	4.33	0.28	1.21
		Acceptance Total	**4.32**
		Acceptability %	**86.32%**

Table 5.3: acceptability % with question weight factored in

Figures 5.1 to 5.4 below illustrates the distribution of the responses to the four acceptability questions. Interestingly for all four graphs, there are firstly only answers for 3, 4 and 5 from the Likert-scale, and secondly, most of the answers are concentrated on 4's and 5's.

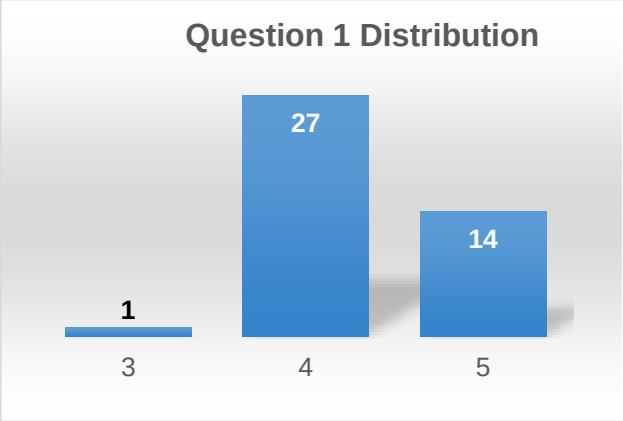


Figure 5.1: Was the UI intuitive and easy to navigate?

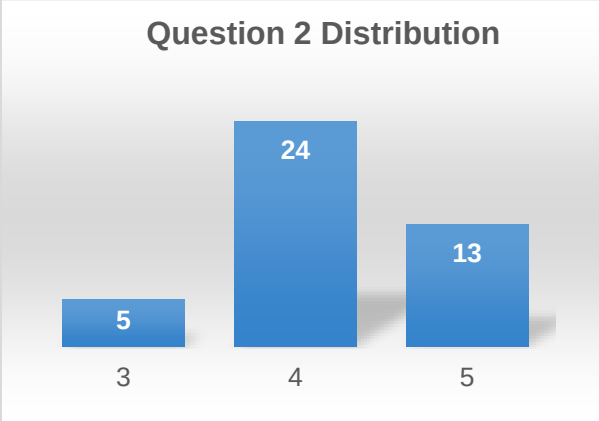


Figure 5.2: Would you recommend the AAC?

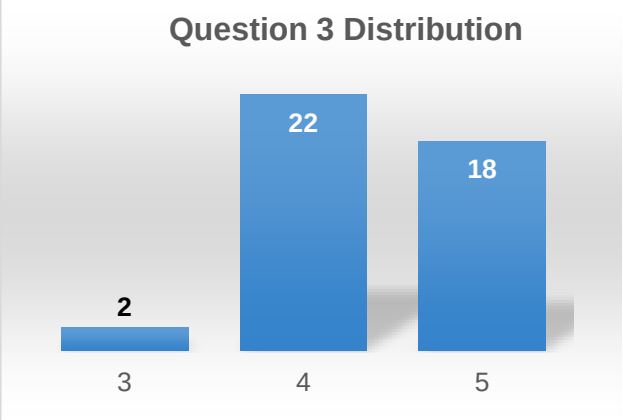


Figure 5.3: Were the results accurate?

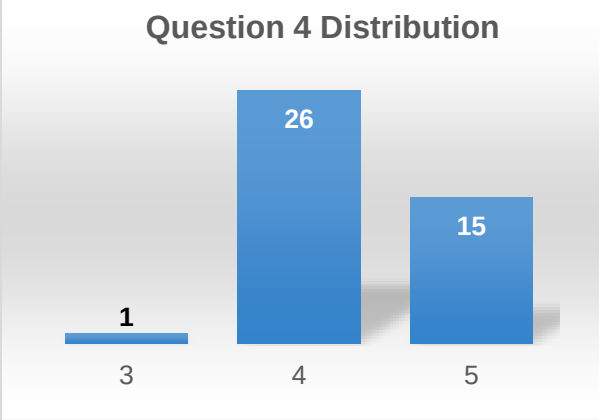


Figure 5.4: Were the results of the AAC assessment tool useful?

Concerning using the mean for central tendency, and not the median, it is clear that with the large representation on 5's in all the graphs the mean had to be used to ensure a more accurate outcome.

5.5 Skewness

All of the skew values lay between -0.5 and 0.5, and, as indicated in section 4.7, this distribution can be viewed as supporting a normal distribution. From the graphs above, it is clear that it is not a perfect normal distribution in any of the cases, but the charts are not seriously skewed.

5.6 Kurtosis

All of the kurtosis values indicate a platykurtic distribution, meaning they all have negative values. This suggests a tendency towards flatness on the graph. From figures 5.1 – 5.4, it is clear that values are concentrated mainly on four and five, and that may lead to a negative tendency.

An average kurtosis of -0.69 is again evidence of not an utterly normal distribution, but this does not, however, influence the reliability of the responses.

5.7 Standard Deviation

The standard deviations (σ) for the four questions reflect what can be seen on the graphs as well. These meagre numbers indicate that most data points are very close to the mean. All data points are one Likert number from the mean.

For two standard deviations from the mean, 98% of the data points are covered in questions one, two and four. Question 3 comprises 88% as there are five selections of 3 that are not included in the range. Three standard deviations from the mean 100% of the data points are included for all four the questions.

Based on this analysis it is clear that the standard deviation values for AAC acceptability are tightly dispersed from the mean, and an assumption can be made that there is general trustworthiness in the selections from the respondents.

5.8 Discussion and Summary

This chapter has taken an in-depth look at the survey results to test the acceptability of the AAC tool. Following the deductive approach, the chapter aimed to verify or falsify the hypothesis that the AAC can be valuable, useful, accurate and therefore acceptable to the Agile community.

Some sub-research questions listed in Table 5.4 below had to be answered before the AAC survey were launched:

Research Question	**Answered?**

a. What are the general unique list of Agile adoption challenges and success factors in literature?	This question was answered in sections 2.4 and 2.5.
b. What is the general priority of the various Agile adoption challenges in software development industries?	Answered in Chapter 4 (Refer section 4.8).
c. What is the relationship between the various Agile adoption challenges originating areas?	It is answered in section 2.4.3.1.
d. Are all industries, doing software development, similarly impacted by Agile adoption challenges?	See section 2.4.4. No definitive mapping between industries and challenges could be found in this research.
e. What are the various other variables describing Agile adoption challenges in Agile software development? The answer to this question will assist in building a comprehensive filter for the automated Agile adoption coach.	It is answered in section 2.4.3.2. Also, see section 2.4.4.
f. What is the mapping between Agile adoption challenges and recorded possible solutions to these challenges in the Agile software development space?	Answered in section 2.6

Table 5.4: Sub research questions answered

Besides the above, the actual AAC, together with the survey, subsequently had to be built. It took as input all of the above answers to the research questions stated.

A pilot was executed to enhance the construct reliability, and it contributed significantly towards the enhancement of the AAC as well as the survey.

The final result in Table 5.3, showing the weighted acceptability score of 86%, is far higher than the target score of 70% (which was the benchmark established in section 3.7).

The respondents' feedback, in general, were quite favourable for all four questions. The results on the question for recommending the AAC tool was just a little lower than the other responses.

Concerning the internal consistency reliability, the Cronbach's α method was used. It validated the internal consistency with a value larger than 0.7 for the responses on the AAC acceptability survey.

There are currently various Agile maturity assessments on the Internet (Linders, 2014). The AAC is not like any of them. Some are entirely free; some are free for one user and others you need to register and pay for the use of the tool. Most of these assessments are based on the Agile principles, Scrum and Extreme Programming practices and then test the output against specific dimensions. For example, culture, planning, requirements and teamwork, to name a few (Cohn et al., 2019). Almost all of these assessments would show a meagre score for companies just starting off on their Agile journey. These companies would not know where to start as nearly everything would be shown as broken. The AAC will help companies to kickstart their Agile journey by helping them to find the priority of challenges to focus on and then also to know what possible solutions there are for these challenges.

As there were only two respondents from outside of South Africa, a point of contention is that this survey is not representative of the global Agile community. There is, however, a sufficient representation of the South African Agile community. They were quite supportive in both the priority and AAC acceptability surveys.

Chapter 6: **CONCLUSION**

6.1 Summary of findings

The end goal and main research question and hypothesis in this research have been analysed in chapter 5, and from that, a very high positive AAC acceptability percentage (86.32%) was attained.

Primary research question: *"Will an automated agile adoption framework be accurate, acceptable and useful to Agile coaches, helping companies adopting Agile software development?"*

Following the deductive approach, we cannot, unfortunately, deduce or generalise with confidence that the AAC would *globally* be accepted in the same manner as in the South African context (as mentioned above, there was only a 5% response rate from international Agile respondents). We can, however, generalise in stating that there is a fair chance that the AAC will be accepted in the South African context, as 95% of the responses hail from this country. The hypothesis has, therefore, not been satisfied fully but sufficiently enough for the South African context.

All the research subquestions, which were answered in previous chapters and are summarised in Table 5.4 in chapter 5, were prerequisites to being completed before the main research question could be addressed.

From this research subquestions, the main results are summarised in Table 6.1 below:

Sub Research Question	Main result
a. What are the general unique list of Agile adoption challenges and success factors in literature?	(a) A normalised list of 97 Agile adoption challenges was extrapolated from research. (b) A unique list of 143 Agile adoption success factors and possible solutions was extracted through a literature study survey. The hypothesis stated in 2.1 has, therefore, also been validated.
b. What is the general priority of the various Agile adoption challenges in software development industries?	Through a quantitative survey, the severity of the impact of Agile adoption challenges, as well as the difficulty in resolving them, were collected on the 97 challenges. This

	formed the base to calculate the priority of each of the 97 challenges. The hypothesis stated in 3.5.2 can now also be validated in part as most of the respondents came from the South African context. One would not be able to say without a doubt that the priorities would have been the same if a more significant global group participated.
c. What is the relationship between the various Agile adoption challenges originating areas?	Seven root areas where Agile adoption challenges originate from were found in the literature.
d. Are all industries, doing software development, similarly impacted by Agile adoption challenges?	The literature surveyed were not conclusive on this research question, and therefore, there was no definitive finding in this regard. This subject might be a candidate for a further quantitative study.
e. What are the various other variables describing Agile adoption challenges in Agile software development? The answer to this question will assist in building a comprehensive filter for the automated Agile adoption coach.	Besides the originating areas, we found in literature 9 Agile challenges categories. This is another grouping that combines similar challenges. These categories were mapped to all the challenges.
f. What is the mapping between Agile adoption challenges and recorded possible solutions to these challenges in the Agile software development space?	Mainly through a literature survey as well as the researcher's subjective input, the mapping between challenges and possible solutions were done. The results are listed in section 2.5.

Table 6.1: Main sub-question results

6.2 Conclusions

6.2.1 Implications

The significance of the study has been partly addressed at the end of the literature review in section 2.6 talking to the theoretical contribution and importance on the findings of the Agile adoption challenges and success factors literature review. The framework developed provides currently quite an exhaustive list on the two main attributes supporting the AAC. This is the first time that such lists have been generated from existing literature. These lists and the priority of the challenges will be updated regularly in order to keep the framework relevant. The impact of this study could be far-reaching for both Agile coaches and organisations where the AAC are used.

Utilising the AAC, Agile coaches could now have the means to show value to the customer much faster. This perceived return on investment would increase the value perception of the coach from the organisation, and may, in return, lead to more work from that organisation (Noruwana, 2010:69). If the coach is a consultant, the benefit could be far-reaching for his/her career in opening up other opportunities at other organisations. The immediate value is not in resolving challenges, but in identifying Agile adoption challenges and possible solutions.

The potential implications for organisations are:

- The potential to transition faster as the solution identification phase is dramatically shortened.
- The potential to save money on Agile coaching as the transition process should be shortened and the integrity of the findings should be higher.
- The potential to enjoy the benefits of productivity enhancements and other Agile mindset benefits should be quicker.

6.2.2 Consistency & Inconsistencies

The literature reviewed is inconsistent in its approach to Agile adoption. Section 2.6 elaborates on the four approaches towards Agile adoption. This research does not attempt to articulate a holistic Agile adoption framework, but it does support the notion that for a true adoption to occur the focus needs to be on the Agile values and principles (Conboy et al., 2010:50). Measuring, for example, Agile adoption based on practices adoption and method adherence would lead to a sub-standard adoption according to Dikert, Paasivaara and Lassenius (2016) and Pitkänen (2015). Very little research has

been conducted discussing the challenges of adopting the four values and 12 principles of Agile (Conboy et al., 2010:49; Gandomani et al., 2013:622; Hajjdiab & Taleb, 2011a:8-9; Dikert, Paasivaara & Lassenius, 2016). Gandomani and Nafchi (2015), Pikkarainen et al. (2012) and Qumer and Henderson-Sellers (2008) for example, place much emphasis on the practices' adoption rather than the Agile values and principles. Understanding these challenges, Agile adoption would also significantly improve as the focus would be on the correct thing. A few frameworks for adoption have been studied (see Table 2.1), but most of them place much emphasis on the practices. They are further typically quite complex to understand and implement.

The AAC is only a tool that forms part of one of the steps towards an Agile adoption and is therefore aimed to be as simplistic as possible.

In the literature review it emerged that there is very little research (six studies) that combine Agile challenges with suggested solutions (Boehm & Turner, 2005; Scrumology, 2009; Conboy et al., 2010; Hajjdiab & Taleb, 2011b; Gandomani et al., 2013 and Dikert, Paasivaara & Lassenius, 2016). In general, these challenges are also quite limited and cover only a subset of the scope of all Agile challenges.

The results of the literature reviews form an intersection from multiple studies on challenges and success factors, and one can, therefore, state that the findings are consistent with the literature. There is not, however, one specific research study that addresses the same unabridged scope of challenges and success factors of this study.

The findings support the view reflected in the literature review that to understand the Agile adoption landscape, the organisation first has to understand their inherent challenges and secondly need to know how to overcome them (Agarwal & Majumdar, 2012:43). Finally, literature findings also support the need for an automated framework to assist in Agile adoption (Ayed, 2014). As the originating areas and categories have been taken from literature, it is inherently supported.

6.3 Limitations and assumptions

The main research question result is limited due to the sample group size and country scope for the AAC acceptability survey. If the country representation were large enough,

the research would have been able to falsify or validate more confidently the main research question. The snowball technique did not realise the international response as expected.

Specific feedback received from Agile coaches on the AAC acceptability survey is that there is not enough focus on the main Agile adoption challenges and that some challenges may weigh inordinately higher than others. For example, lack of executive commitment, people and culture and neglecting the need for technical excellence are critical challenges to address. This specific limitation will be mitigated in future versions of the AAC.

A limiting factor for the AAC is the fact that challenges could not be grouped successfully into specific industry categories, as no available literature can provide that data comprehensively. This would have helped considerably in the refining process when selecting significant Agile adoption challenges. Conboy (2009:340) suggests there would be significant value if Agile challenges could be mapped to relevant industries as related solutions would be easier to recognise.

6.4 Recommendations for Future Research

The assumption is that the literature would provide most of the necessary data to determine Agile adoption challenges, their impact on organisations and possible solutions to the challenges identified.

Future studies may want to elaborate on the literature review to refine the results by extending the literature review with qualitative research for the first sub-objective. This research may then focus on improving the literature review challenges results by smoothing out possible ambiguities, deleting redundancies that are not obvious and adding possible oversights.

The sixth sub-objective could lend itself to a qualitative study to get the best results. Qualitative research could be much more focused on the experiences of Agile coaches. Coaches could provide information on what solutions generally work best with specific Agile adoption challenges.

On the fourth sub-question, it would be an interesting quantitative study to determine whether or not there is a link between Agile adoption challenges and various industries. This study could not definitively prove through the literature review whether such a relationship exists or not. At this point, the researcher believes there is a link, but there might be considerable overlap between the industries.

In support of such a future study concerning the current research, Kitchenham et al. (2002:23) address a challenge with surveys in that it is more exploratory than looking at causal relationships. Also, Miles and Huberman (1994:16) indicated that research is becoming more and more complex and increasingly multisite, multi-method studies are introduced, which utilises combinations of qualitative and quantitative inquiry. Unfortunately, this research would have breached its time and scope restrictions if it included all the suggested research shortcomings.

Morgan (2007:73) stipulates that a pragmatic approach not only supports a combined quantitative and qualitative approach but act as a foundation to refocus the area of social science research in the direction the researcher favours. This was also the approach of this research to address apparent gaps in the Agile adoption field and to come up with something genuinely beneficial, which might benefit the holistic Agile community.

7. REFERENCES

Abawi, K. 2008. Qualitative and Quantitative Research. In *World Health Organization/Geneva Foundation for Medical Education and Research*. Kabul: Ministry of Public Health Kabul.

Abrahamsson, P., Warstab, J., Siponen, M.T. & Ronkainen, J. 2003. New Directions on Agile Methods: A Comparative Analysis. In *International Conference on Software Engineering*. Portland, Oregon: Technical Research Centre of Finland, VTT Electronics.

Adkins, L. 2010. *Coaching Agile Teams: A Companion for ScrumMasters, Agile Coaches, and Project Managers in Transition*. Addison-Wesley.

Agarwal, M. & Majumdar, R. 2012. Reducing Complexity Faced By Large Organizations Using Distributed Scrum in Agile Development. 1(1):35–43.

Agile Alliance. 2001. *Guide to Agile practices*. Available: https://www.agilealliance.org/agile101/guide-to-agile/ [2016, May 30].

Alam, S.S. & Chandra, S. 2014. Agile Software Development : Novel Approaches For Software Engineering. 36–40.

Almeida, F. 2017. Challenges in Migration from Waterfall to Agile Environments. *World Journal of Computer Application and Technology*. 5(3):39–49. DOI: 10.13189/wjcat.2017.050302.

Anderson, D.R., Sweeney, D.J., Williams, T.A., Freeman, J. & Shoesmith, E. 2014. *Statistics for Business and Economics*. Third ed. Andover: Cengage Learning.

Arppe, A. 2008. Univariate, bivariate, and multivariate methods in corpus-based lexicography – a study of synonymy. In *Corpus Linguistics Conference*. Helsinki: Helsinki University Print. 600.

Ayed, H. 2014. Supported approach for agile methods adaptation : An adoption study. In *RCose 2014*. Hyderabad, India. DOI: 10.1145/2593812.2593820.

Babbie, E. & Mouton, J. 2012. *The Practice of Social Research*. Cape Town: Oxford University Press Southern Africa.

Bavani, R. 2009. Critical Success Factors in Distributed Agile for Outsourced Product Development. *Proceedings of CONSEG-09:*. 5. DOI: 10.1111/1468-2370.00040.

Beck, K., Martin, R., Cockburn, A., Fowler, M. & Highsmith, J. 2001. *Agile Manifesto*. Available: http://www.agilemanifesto.org [2016, May 30].

Benefield, R. 2010. Seven dimensions of agile maturity in the global enterprise: A case study. *Proceedings of the Annual Hawaii International Conference on System Sciences*. 1–7. DOI: 10.1109/HICSS.2010.337.

Black, K. 2010. *Business Statistics: Contemporary Decision Making*. St. Louis: John Wiley & Sons.

Bloomfield, L. 2015. Lightweight Interventions for Reflective Documentation. Carnegie Mellon University.

Boehm, B. & Turner, R. 2005. Management challenges to implementing agile processes in traditional development organizations. *IEEE Software*. 22(5):30–39. DOI: 10.1109/MS.2005.129.

Brown, D.D. 2013. Common Success Factors. In *Agile User Experience Design*. Waltham: Elsevier Inc. 121–146. DOI: 10.1016/B978-0-12-415953-2.00004-2.

Bryman, A. 2006. Integrating quantitative and qualitative research: How is it done? *Qualitative Research*. 6(1):97–113.

Burrell, G. & Morgan, G. 1979. Sociological Paradigms and organisational Analysis - Elements of the Sociology of Corporate Life. *Sociological Paradigms and organisational analysis*. 448. DOI: 10.1177/003803858001400219.

CAMP. 2011. *Researching Forced Migration in Pakistan: An Introduction to Research Ethics, Quantitative & Qualitative Methods*. Islamabad: CAMP.

Chan, F. & Thong, J. 2009. Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems*. 46(4):803–814. DOI: 10.1016/j.dss.2008.11.009.

Chen, R. (Ronxin), Ravichandar, R. & Proctor, D. 2016. Managing the transition to the new agile business and product development model: Lessons from Cisco Systems. *Business Horizons*. 59(6):635–644. DOI: 10.1016/j.bushor.2016.06.005.

Chiu, J.M. & Liu, W.L. 2008. A Study of the Feasibility of Network Tutorial System in Taiwan. *Educational Technology & Society*. 11(1):208–225.

Chow, T. & Cao, D.-B. 2008. A survey study of critical success factors in agile software projects. *Journal of Systems and Software*. 81(6):961–971. DOI: 10.1016/j.jss.2007.08.020.

Cohen, L. & Manion, L. 1980. *Research Methods in Education*. London: Groom Helm Ltd.

Cohn, M. 2005. *Agile Estimating and Planning*.

Cohn, M. 2009. *Succeeding with Agile Software development using scrum*. Pearson Education.

Cohn, M. & Ford, D. 2003. Introducing an agile process to an organization. *Computer*. 36(6):74–78. DOI: 10.1109/MC.2003.1204378.

Cohn, M., Kasperowski, R., Edmondson, A. & Wolff, S. 2019. *The World's Largest Agility Assessment and Continuous Improvement platform*. Available: https://www.comparativeagility.com/ [2019, August 12].

Conboy, K. 2009. Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research*. 20(3):329–354. DOI: 10.1287/isre.1090.0236.

Conboy, K., Coyle, S., Wang, X. & Pikkarainen, M. 2010. People Over Process: Key People Challenges in Agile Development. *IEEE Software*. 28(4):48–57. DOI: 10.1109/MS.2010.132.

Conforto, E.C., Salum, F., Amaral, D.C., da Silva, S.L. & de Almeida, L.F.M. 2014. Can Agile Project Management be Adopted by Industries Other than Software Development? *Project Management Journal*. 45(3):21–34. DOI: 10.1002/pmj.21410.

Corbucci, H., Goldman, A., Katayama, E., Kon, F., Melo, C. & Santos, V. 2011. Genesis and Evolution of the Agile Movement in Brazil--Perspective from Academia and Industry. *Software Engineering (SBES), 2011 25th Brazilian Symposium on*. 98–107.

Creswell, J.W. 2003. *Research Design. Qualitative, Quantitative and Mixed Methods*

Approaches. 2nd ed. Thousand Oaks: SAGE Publications.

Cyfar. 2018. *Inferential Analysis*. Available: https://cyfar.org/inferential-analysis [2018, May 13].

Dawson, C. 2008. Pratical research methods: A user-friendly guide to mastering research. *Vasa*. 153. DOI: 10.1017/CBO9781107415324.004.

Denning, S. 2015. How to make the whole organization Agile. *Strategy & Leadership*. 43(6):10–17. DOI: 10.1108/SL-09-2015-0074.

Denning, S. 2016. *Should We Change the Agile Manifesto?* Available: https://www.agilealliance.org/should-we-change-the-agile-manifesto/ [2018, March 23].

Dikert, K., Paasivaara, M. & Lassenius, C. 2016. Challenges and success factors for large-scale agile transformations: A systematic literature review. *Journal of Systems and Software*. 119:87–108. DOI: 10.1016/j.jss.2016.06.013.

Dingsøyr, T. & Moe, N.B. 2013. Research challenges in large-scale agile software development. *ACM SIGSOFT Software Engineering Notes*. 38(5):38. DOI: 10.1145/2507288.2507322.

Dodge, Y. 2003. *The Oxford Dictionary of Statistical Terms*. Oxford: Oxford University Press.

Easterby-Smith, M., Thorpe, R., Jackson, P. & Lowe, A. 2012. *Management Research*. 4th ed. London: SAGE Publications.

Fitzgerald, B. & Stol, K.-J. 2014. Continuous software engineering and beyond: trends and challenges. *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering - RCoSE 2014*. 1–9. DOI: 10.1145/2593812.2593813.

Fraenkel, J.R. & Wallen, N.E. 2013. How to design and evaluate research in education. *Journal of Chemical Information and Modeling*. 53(9):1689–1699. DOI: 10.1017/CBO9781107415324.004.

Gandomani, T.J. & Nafchi, M.Z. 2015. An empirically-developed framework for Agile transition and adoption: A Grounded Theory approach. *Journal of Systems and Software*. 107:204–219. DOI: 10.1016/j.jss.2015.06.006.

Gandomani, T.J., Zulzalil, H., Ghani, A.A.A., Sultan, A.B.M. & Nafchi, M.Z. 2013. Obstacles in moving to agile software development methods; At a Glance. *Journal of Computer Science*. 9(5):620–625. DOI: 10.3844/jcssp.2013.620.625.

Gonçalves, L. 2016. *5 Crazy Reasons Why Agile does not Work In Germany*. Available: http://lmsgoncalves.com/2016/05/01/5-crazy-reasons-agile-not-work-germany/ [2016, May 30].

GoodData. 2018. *Normality Testing - Skewness and Kurtosis*. Available: https://help.gooddata.com/display/doc/Normality+Testing+-+Skewness+and+Kurtosis [2018, December 17].

Gorans, P. & Kruchten, P. 2016. A Guide to Critical Success Factors in Agile Delivery. In *Agile and Beyond*. IBM Corporation. Available: http://www.businessofgovernment.org/report/guide-critical-success-factors-agile-delivery.

Grademiners. 2015. *How to Write Chapter 3 of the Dissertation*. Available:

https://grademiners.com/blog/how-to-write-chapter-3-of-the-dissertation [2018, April 07].

Guckenheimer, S. & Loje, N. 2012. *Agile Software Engineering with Visual Studio: from concept to continuous feedback*. second ed. Boston: Addison-Wesley Professional.

De Haaff, B. 2017. *How to Determine the Right Agile Development Methodology*. Available: https://www.huffingtonpost.com/brian-de-haaff/how-to-determine-the-righ_b_9567344.html [2018, January 06].

Hajjdiab, H. & Taleb, A.S. 2011a. Adopting Agile Software Development: Issues and Challenges. *International Journal of Managing Value and Supply Chains*. 2(3):1–10. DOI: 10.5121/ijmvsc.2011.2301.

Hajjdiab, H. & Taleb, A.S. 2011b. Agile adoption experience: A case study in the U.A.E. *ICSESS 2011 - Proceedings: 2011 IEEE 2nd International Conference on Software Engineering and Service Science*. 31–34. DOI: 10.1109/ICSESS.2011.5982247.

Hakim, C. 2000. *Research Design: Successful designs for social and economic research*. 2nd ed. Abingdon: Routledge. Available: http://samples.sainsburysebooks.co.uk/9781135125653_sample_547151.pdf.

Hinkelmann, K. & Witschel, H.F. 2013. DOI: 10.1016/j.molcel.2009.09.013.

Hoogendoorn, S. 2012. *Evolving Agile*. Available: https://www.capgemini.com/2012/04/evolving-agile/ [2018, December 08].

Hunt, J. 2006. *Agile Software Construction*. V. 49. London: Springer London. DOI: 10.1007/1-84628-262-4.

De Jager, K. & Steele, D. 2015. *Harvard Uct : Handbook on Citation*. Cape Town: University of Cape Town.

Kairi, T. 2011. Coaching agile software development teams : A case study from mentoring viewpoint. University of Helsinki.

Kitchenham, B.A. & Jones, L. 1997. Evaluating Software Engineering Methods and Tool part 5: the Influence of Human Factors. *ACM SIGSOFT Software Engineering Notes*. 22(1):13–15.

Kitchenham, B.A., Pfleeger, S.L., Pickard, L.M., Jones, P.W., Hoaglin, D.C., Emam, K. El & Rosenberg, J. 2002. Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*. 28(8):721–734. DOI: 10.1109/TSE.2002.1027796.

Koziy, Y. 2017. Going Agile: build the right thing, execute faster than your competition. In *RockIt Digital Conference*. Chisinau. Available: https://www.slideshare.net/yurkoziy/going-agile-build-the-right-thing-execute-faster-than-your-competition.

Kropp, Martin & Meier, A. 2015. Agile Success Factors. *Clinical and experimental rheumatology*. 32 Suppl 8(2):1–4. Available: http://www.ncbi.nlm.nih.gov/pubmed/24743859.

Laanti, M., Salo, O. & Abrahamsson, P. 2011. Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology*. 53(3):276–290. DOI: 10.1016/j.infsof.2010.11.010.

Leedy, P.D., Omrod, J.E. & Leedy, P.D. 2013. *Practical Research: Planning and design*.

10th ed. New Jersey: Pearson Merill Prentice Hall.

Levy, D. 2009. *Agile software development*. Available: http://www.gatherspace.com/static/agile_software_development.html [2016, May 30].

Linders, B. 2014. *Agile Self-assessments*. Available: http://www.benlinders.com/agile-self-assessments [2019, August 12].

Lindstrom, L. & Jeffries, R. 2004. Extreme programming and agile software development methodologies. *Information Systems Management*. 21:41–52. DOI: 10.1201/1078/44432.21.3.20040601/82476.7.

Lindvall, M., Muthig, D., Dagnino, A., Wallin, C., Stupperich, M., Kiefer, D., May, J. & Kahkonen, T. 2004. Agile software development in large organizations. *Computer*. 37(12):26–34. DOI: 10.1109/MC.2004.231.

MacMillan, J.H. & Schumacher, S. 2001. *Descriptive statistics. Research in Education: A Conceptual Introduction*. 5th ed. New York: Longman.

Marchenko, A. & Abrahamsson, P. 2008. Scrum in a multiproject environment: An ethnographically-inspired case study on the adoption challenges. *Proceedings - Agile 2008 Conference*. 15–26. DOI: 10.1109/Agile.2008.77.

Mbuya, P.D. 2016. Agile and conventional methodologies: An empirical investigation of their impact on software quality parameters. (January):122.

McCluskey, A. & Lalkhen, A.G. 2007. Statistics II: Central tendency and spread of data. *Continuing Education in Anaesthesia, Critical Care and Pain*. 7(4):127–130. DOI: 10.1093/bjaceaccp/mkm020.

Merriam-webster. 2018. *Definition of useful for Students*. Available: https://www.merriam-webster.com/dictionary/useful [2018, June 09].

Miles, M. & Huberman, A. 1994. *Qualitative Data Analysis; An Expanded Sourcebook*. 2nd ed. SAGE Publications.

Milne, J. 1999. *Questionnaires: Advantages and Disadvantages*. Available: http://www.icbl.hw.ac.uk/ltdi/cookbook/info_questionnaires/ [2018, April 29].

Mishra, D. & Mishra, A. 2011. Pattern detection for conceptual schema recovery in data-intensive systems. *Journal of Software Maintenance and Evolution: Research and Practice*. 23(8):549–564.

Misra, S.C., Kumar, U., Kumar, V. & Grant, G. 2006. The organizational changes required and the challenges involved in adopting agile methodologies in traditional software development organizations. In *2006 1st International Conference on Digital Information Management, ICDIM*. 25–28.

Misra, S.C., Kumar, V. & Kumar, U. 2009. Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*. 82(11):1869–1890. DOI: 10.1016/j.jss.2009.05.052.

Mitchell, V. 1996. Assessing the reliability and validity of questionnaires: an empirical example. *Journal of Applied Management Studies*. (5):199–208.

Morgan, D.L. 2007. Paradigms Lost and Pragmatism Regained. *Journal of Mixed Methods Research*. 1(1):48–76. DOI: 10.1177/2345678906292462.

Mouton, J. 2005. *How to succeed in your Masters and Doctoral Studies*. 9th ed. Pretoria:

Van Schaik.

Nerur, S., Mahapatra, R. & Mangalaraj, G. 2005. Challenges of migrating to agile methodologies. *Communications of the ACM*. 48(5):72–78. DOI: 10.1145/1060710.1060712.

Nguyen, D.S. 2016. Success Factors That Influence Agile Software Development Project Success. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*. 17(1):172–222.

Noruwana, N. 2010. How do South African organisations adopt agile methodologies for software development in practice? Available: https://open.uct.ac.za/handle/11427/11708.

Noruwana, N. & Tanner, M. 2012. Understanding the structured processes followed by organisations prior to engaging in agile processes: A South African Perspective. *South African Computer Journal*. 48(48):41–58. Available: http://reference.sabinet.co.za/sa_epublication_article/comp_v48_a5.

O'Connor, R. & Duchonova, N. 2014. Assessing the Value of an Agile Coach in Agile Method Adoption. *Communications in Computer and Information Science*. 425(June):135–146. DOI: 10.1007/978-3-662-43896-1_12.

Oates, B.J. 2006. *Researching Information Systems and Computing*. London: SAGE Publications. DOI: 10.1016/j.ijinfomgt.2006.07.009.

Omrod, J.E. & Leedy, P.D. 2013. *Practical Research: Planning and design*. 10th ed. New Jersey: Pearson Merill Prentice Hall.

ORiordan, D. 2017. *WHEN AGILE GETS DANGEROUS*. Available: https://www.darraghoriordan.com/2017/09/14/agile-gets-dangerous/ [2018, January 04].

Patton, M.Q. 2002. *Qualitative Research and Evaluation Methods*. 3rd ed. Thousand Oaks, California: Sage.

Pikkarainen, M., Salo, O., Kuusela, R. & Abrahamsson, P. 2012. Strengths and barriers behind the successful agile deployment — insights from the three software intensive companies in Finland. *Empirical Software Engineering*. 17:675–702. DOI: 10.1007/s10664-011-9185-5.

Pitkänen, A. 2015. Agile Transformation: A case study. Aalto University. Available: http://urn.fi/URN:NBN:fi:aalto-201512165681.

Prokhorenko, S. 2012. Skiing and boxing: Coaching product and enterprise teams. *Proceedings - 2012 Agile Conference, Agile 2012*. 191–196. DOI: 10.1109/Agile.2012.30.

Qumer, A. & Henderson-Sellers, B. 2008. A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of Systems and Software*. 81(11):1899–1919. DOI: 10.1016/j.jss.2007.12.806.

Qumer, A., Henderson-Sellers, B. & McBride, T. 2007. Agile adoption and improvement model. *Proceedings of the European and Mediterranean Conference on Information Systems, EMCIS 2007*. 2007:21–29. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-77955464821&partnerID=tZOtx3y1.

Ramesh, B., Cao, L. & Baskerville, R. 2010. Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal*. 20(5):449–480. DOI: 10.1111/j.1365-2575.2007.00259.x.

Rodríguez, P., Markkula, J., Oivo, M. & Turula, K. 2012. Survey on agile and lean usage in Finnish software industry. *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement - ESEM '12*. 139. DOI: 10.1145/2372251.2372275.

Sahota, M. 2012. *An Agile Adoption and Transformation Survival Guide: Working with Organizational Culture*.

Saunders, M., Lewis, P. & Thornhill, A. 2003. *Research Methods for business Students*. 3rd ed. Harrowsmith: Prentice Hall.

Saunders, M., Lewis, P. & Thornhill, A. 2016. *Research Methods For Business Students*. 7th ed. V. 53. Pearson. DOI: 10.1017/CBO9781107415324.004.

Schwaber, K. & Sutherland, J. 2017. The Scrum Guide. *Scrum.Org and ScrumInc*. (July):19. DOI: 10.1053/j.jrn.2009.08.012.

Schwaber, K., Laganza, G. & D'Silva, D. 2007. *The Truth about Agile Processes: Frank Answers to Frequently Asked Questions*. Available: https://www.forrester.com/report/The+Truth+About+Agile+Processes/-/E-RES41836.

Scrumology. 2009. *The Top 13 Organization Challenges of Agile Development – and a Solution to Each*. Available: http://s3.amazonaws.com/public.kanemar.com/13OrganizationChallengesofAgile.pdf [2017, December 14].

Shareen, N. 2010. Parental involvement in the education of learners on farm schools in the Citrusdal area. University of the Western Cape.

Sidky, A. 2007. A Structured Approach to Adopting Agile Practices : The Agile Adoption Framework. Virginia Polytechnic Institute and State University.

Sidky, A. & Arthur, J. 2007a. A Disciplined Approach to Adopting Agile Practices : The Agile Adoption Framework. *Innovations in Systems and Software Engineering*. 3(3):203–216.

Sidky, A. & Arthur, J.D. 2007b. A Structured Approach to Adopting Agile Practices : The Agile Adoption Framework A Structured Approach to Adopting Agile Practices : The Agile Adoption Framework. *Innovations in Systems and Software Engineering*. (1).

Sintes, J. 2017. *Integrating UX into an Agile Environment*. Available: https://www.userzoom.com/agile-ux-research-and-testing/integrating-ux-into-an-agile-environment/ [2017, January 06].

SmartSurvey. 2018. *10 Advantages of Online Surveys*. Available: https://www.smartsurvey.co.uk/articles/10-advantages-of-online-surveys [2018, April 29].

Soundararajan, S. & Arthur, J.D. 2011. A Structured Framework for Assessing the. 14–23. DOI: 10.1109/ECBS.2011.26.

Speicher, M., Both, A. & Gaedke, M. 2015. Inuit: The Interface Usability Instrument. In *Marcus A. (eds) Design, User Experience, and Usability: Design Discourse. DUXU 2015. Lecture Notes in Computer Science*. V. 9186. Springer, Cham. DOI: https://doi.org/10.1007/978-3-319-20886-2_25.

Srinivasan, J. & Lundqvist, K. 2009. Using Agile Methods in Software Product Development: A Case Study. *2009 Sixth International Conference on Information Technology: New Generations*. 1415–1420. DOI: 10.1109/ITNG.2009.334.

Srinivasan, J. & Lundqvist, K. 2010. Agile in India : Challenges and Lessons Learned. *Proceedings of the 3rd India software engineering conference (ISEC)*. 125–130. DOI: 10.1145/1730874.1730898.

Statistics How To. 2018. *Kurtosis: Definition, Leptokurtic, Platykurtic*. Available: https://www.statisticshowto.datasciencecentral.com/probability-and-statistics/statistics-definitions/kurtosis-leptokurtic-platykurtic/ [2018, December 18].

Suddaby, R. 2006. What grounded theory is not. *Academy of Management Journal*. 49(4):633–643.

Sukamolson, S. 2005. Fundamentals of quantitative research. *e-Journal for researching teachers*. 2(1):20. DOI: 9781848608641.

Tennent, J. 2013. *The Economist Numbers Guide: The Essentials of Business Numeracy*. 6th ed. London: Profile Books.

Terre Blanche, M., Durrheim, K. & Painter, D. 2007. *Research in Practice*. Cape Town: UCT Press.

Tripp, J.F., Saltz, J. & Turk, D. 2018. Thoughts on Current and Future Research on Agile and Lean : Ensuring Relevance and Rigor. 9:5465–5472.

Vandersluis, C. 2014. Apply agile methodology to non-software enterprise projects. In *PMI Global Congress Proceedings*. 1–7. Available: http://www.pmi.org/learning/library/apply-agile-methodology-nonsoftware-enterprise-projects-9273.

VersionOne. 2015. *The 10th annual state of Agile report*. Available: http://versionone.com/pdf/VersionOne-10th-Annual-State-of-Agile-Report.pdf.

VersionOne. 2016. *The 11th annual state of Agile report*. Available: https://explore.versionone.com/state-of-agile/versionone-11th-annual-state-of-agile-report-2.

Vijayasarathy, L.R. & Turk, D. 2008. Agile software development: A survey of early adopters. *Journal of Information Technology Management*. XIX(2):1–8. DOI: 10.1287/mnsc.48.8.1008.163.

Wang, X., Conboy, K. & Pikkarainen, M. 2012. Assimilation of Agile Practices in Use. *Information Systems Journal*. 22(6):435–455.

Weyer, D. 2016. *5 Crazy Reasons Why Agile May not Work in South Africa*. Available: http://www.leankanban.co.za/2016/05/03/5-crazy-reasons-why-agile-may-not-work-in-south-africa/ [2016, May 30].

Wikipedia. 2018. *Posivitism*. Available: https://en.wikipedia.org/wiki/Positivism [2018, December 26].

8. APPENDIX

8.1 Results from literature survey

Success Factors / Solutions	Reference	Type
Agile is asking for end-to-end working software at the end of every increment. All non-functional requirements are inclusive of that. Significant rework can be introduced later on if it is not addressed early enough.	(Almeida, 2017)	Solution
In large-scale projects or programs, product owners need to collaborate a lot, and the uses of a scrum of scrums for product owners as well as an overarching Product owner can help with this challenge.	(Almeida, 2017)	Solution
Investigate to see what the absolute mandatory legislative is and audit requirements. Agree with management on only making provision for that in the customised method.	(Almeida, 2017)	Solution
Utilising the correct tools traceability in Agile among defects, user stories, test cases, and how it lands up in production should not be difficult. It is essential the whole team understands how to map the work items and the need for it.	(Almeida, 2017)	Solution
Apply throughput accounting rather than cost accounting in software development projects	(Anderson, 2003)	Solution
Assess Internal Quality	(Bavani, 2009)	Success Factor
Compliment People to Improve Processes	(Bavani, 2009)	Success Factor
Cut Communication Loops	(Bavani, 2009)	Success Factor
Ensure Explicit Delegation and Validate Assumptions	(Bavani, 2009)	Success Factor
Facilitate Tool Driven Query Resolution	(Bavani, 2009)	Success Factor
Initiate Test Drives	(Bavani, 2009)	Success Factor
Invest in Root Cause Analysis	(Bavani, 2009)	Success Factor
Manage Effort Variance Constructively	(Bavani, 2009)	Success Factor
Set up the Base Camp!	(Bavani, 2009)	Success Factor
Take Stock of User Stories for Status Checks	(Bavani, 2009)	Success Factor
Explain that Agile values people and that it is part of the core values. Agile transformation is about transforming into a new mindset that can benefit the organisation and individuals.	(Beck et al., 2001)	Solution
Maintenance on multiple releases may increase, but two things should be kept in mind: Agile in itself demands high-quality deliveries. Otherwise, multiple releases will not be sustainable and secondly satisfying the customer through early, and continuous releases are core in Agile and should be a key motivator for the team.		Solution
Address HR issues when you begin your pilot project	(Boehm & Turner, 2005)	Solution
Apply throughput accounting rather than cost accounting in software development projects	(Boehm & Turner, 2005)	Solution
Build up processes rather than tailoring them down.	(Boehm & Turner, 2005)	Solution
Conduct empirical studies of which classes of change are more unpredictable and therefore suited for agile methods	(Boehm & Turner, 2005)	Solution

Define specific functionality or responsibilities that you are going to address with agile approaches.	(Boehm & Turner, 2005)	Solution
Develop architectures that support the compartmentalisation of agile and traditional teams.	(Boehm & Turner, 2005)	Solution
Develop management and architectural practices for hybrid agile and plan-driven methods.	(Boehm & Turner, 2005)	Solution
Do some serious preparation upfront	(Boehm & Turner, 2005)	Solution
align or redefine traditional milestone reviews to better fit an iterative approach	(Boehm & Turner, 2005)	Solution
Educate stakeholders	(Boehm & Turner, 2005)	Solution
Emphasise value.	(Boehm & Turner, 2005)	Solution
Establish guidelines for safe and agility-compatible process maturity assessments	(Boehm & Turner, 2005)	Solution
Evaluating risks is the best overall approach to determining how much agility (or any attribute, for that matter) is enough.	(Boehm & Turner, 2005)	Solution
Identify incompatible assumptions (model clashes) and compatible assumptions (synergies) between agile and traditional methods	(Boehm & Turner, 2005)	Solution
Implement agile practices that support existing processes or new organisational priorities.	(Boehm & Turner, 2005)	Solution
Investigate and update contracting practices to support agile concepts	(Boehm & Turner, 2005)	Solution
Pick good people and reward the results of pilot projects.	(Boehm & Turner, 2005)	Solution
Reorient reward systems to recognise both individual and team contributions.	(Boehm & Turner, 2005)	Solution
Research how to modify or reconceive legacy systems	(Boehm & Turner, 2005)	Solution
Translate agile and software issues into management and customer language	(Boehm & Turner, 2005)	Solution
Understand how communication occurs within development teams	(Boehm & Turner, 2005)	Solution
While the whole team is not fully Agile, and there exist specific interdependencies between various method teams, key upfront agreements and risk acceptance have to be understood and agreed upon. This approach requires much discipline and takes away from the productivities a full Agile team could have achieved.	(Boehm & Turner, 2005)	Solution
Adapt and Evolve	(Brown, 2013)	Success Factor
Communication	(Brown, 2013)	Success Factor
Define the Big Picture	(Brown, 2013)	Success Factor
Project Over Process	(Brown, 2013)	Success Factor
Team Dynamics	(Brown, 2013)	Success Factor
Training	(Brown, 2013)	Success Factor
Appropriate technical training to team	(Chow & Cao, 2008)	Success Factor

Coherent, self-organising teamwork	(Chow & Cao, 2008)	Success Factor
The collocation of the whole team	(Chow & Cao, 2008)	Success Factor
Committed sponsor or manager	(Chow & Cao, 2008)	Success Factor
Cooperative organisational culture instead of hierarchal	(Chow & Cao, 2008)	Success Factor
Correct integration testing	(Chow & Cao, 2008)	Success Factor
Customer having full authority	(Chow & Cao, 2008)	Success Factor
Delivering the most essential features first	(Chow & Cao, 2008)	Success Factor
Facility with the proper agile-style work environment	(Chow & Cao, 2008)	Success Factor
Following agile-oriented configuration management process	(Chow & Cao, 2008)	Success Factor
Following agile-oriented project management process	(Chow & Cao, 2008)	Success Factor
Following agile-oriented requirement management process	(Chow & Cao, 2008)	Success Factor
Good customer relationship	(Chow & Cao, 2008)	Success Factor
The good progress tracking mechanism	(Chow & Cao, 2008)	Solution
Honouring regular working schedule – no overtime	(Chow & Cao, 2008)	Success Factor
Managers knowledgeable in the agile process	(Chow & Cao, 2008)	Success Factor
Managers who have a light-touch or adaptive management style	(Chow & Cao, 2008)	Success Factor
Oral culture placing a high value on face-to-face communication	(Chow & Cao, 2008)	Success Factor
Organisations where the agile methodology is universally accepted	(Chow & Cao, 2008)	Success Factor
Project nature being non-life-critical	(Chow & Cao, 2008)	Success Factor
Project type being of variable scope with the emergent requirement	(Chow & Cao, 2008)	Success Factor
Projects with a dynamic, accelerated schedule	(Chow & Cao, 2008)	Success Factor
Projects with no multiple independent teams	(Chow & Cao, 2008)	Success Factor
Projects with a small team	(Chow & Cao, 2008)	Success Factor
Projects with up-front cost evaluation done	(Chow & Cao, 2008)	Success Factor
Projects with up-front risk analysis done	(Chow & Cao, 2008)	Success Factor
Pursuing simple design	(Chow & Cao, 2008)	Success Factor
Regular delivery of software	(Chow & Cao, 2008)	Success Factor
Reward system appropriate for agile	(Chow & Cao, 2008)	Success Factor
Right amount of documentation	(Chow & Cao, 2008)	Success Factor
Rigorous refactoring activities	(Chow & Cao, 2008)	Success Factor
Strong communication focus with daily face-to-face meetings	(Chow & Cao, 2008)	Success Factor
Strong customer commitment and presence	(Chow & Cao, 2008)	Success Factor
Strong executive support	(Chow & Cao, 2008)	Success Factor
Team members with great motivation	(Chow & Cao, 2008)	Success Factor
Team members with high competence and expertise	(Chow & Cao, 2008)	Success Factor
Well-defined coding standards upfront	(Chow & Cao, 2008)	Success Factor
At the start of the project, it is okay if various teams can agree on the estimation method and also agree on a sizing reference story. E.g. Everyone understands what a user story size of three looks like. They all use it from there on as a reference. This way, all teams will have their velocity but based on the same referencing system.	(Cohn, 2005)	Solution
360° feedback is a must.	(Conboy et al., 2010)	Solution

Assess agility regarding agile values and principles, not practice adherence.	(Conboy et al., 2010)	Solution
Build a sharing and learning environment to empower team decisionmaking.	(Conboy et al., 2010)	Solution
Collecting and sharing successful adoption stories and positive experiences.	(Conboy et al., 2010)	Solution
Combine development and training program to provide customised training materials on social skills, using developers' examples.	(Conboy et al., 2010)	Solution
Customer company runs training sessions on essential topics within the business domain and the company-specific area(s).	(Conboy et al., 2010)	Solution
Dedicated mentor for new staff.	(Conboy et al., 2010)	Solution
Develop specific recruiting practices tailored to agile methods to hire the right people.	(Conboy et al., 2010)	Solution
Encourage task self-assignment to allow a developer to work in different areas and learn new skills.	(Conboy et al., 2010)	Solution
Ensure cross-team observation/validation of agile practices.	(Conboy et al., 2010)	Solution
Ensure multiple members get agile training or attend agile conferences.	(Conboy et al., 2010)	Solution
Feedback outside stand-ups, allowing the documentation of any fears, issues or concerns inappropriate for discussion in open forum	(Conboy et al., 2010)	Solution
Implement a democratic voting system.	(Conboy et al., 2010)	Solution
Mature Agile coaching and championing.	(Conboy et al., 2010)	Solution
Performance evaluation needs to consider the breadth of skills, not just depth.	(Conboy et al., 2010)	Solution
Performance evaluation to apply much higher weighting for mentoring, voluntary contributions, etc.	(Conboy et al., 2010)	Solution
Project manager plays the role of facilitator.	(Conboy et al., 2010)	Solution
Provide small training modules (on a frequent basis), making it interactive to allow developers to acquire niche business knowledge required by the project	(Conboy et al., 2010)	Solution
Put newly recruited graduates on agile projects to get hands-on experience.	(Conboy et al., 2010)	Solution
Recruit staff and graduates with a combination of IT and business knowledge	(Conboy et al., 2010)	Solution
Reintroduce specific roles when it is perceived beneficial to teams with, e.g. large team size, conflicts between developers	(Conboy et al., 2010)	Solution
Stand up meetings voluntary for new junior developers.	(Conboy et al., 2010)	Solution
Try to have multiple 'bought-in' developers on each team.	(Conboy et al., 2010)	Solution
Use pair programming and pair rotation to distribute knowledge and facilitate learning.	(Conboy et al., 2010)	Solution
Use team recruiting to find the right person working in the team.	(Conboy et al., 2010)	Solution
Using proper documentation to back up communication.	(Conboy et al., 2010)	Solution
Weaker developers paired with those who had more experience, taking joint responsibility for requirements.	(Conboy et al., 2010)	Solution

Firstly you need to understand the function of the various mainstream Agile methods currently. You can also evaluate by looking at Project size, Team size, Iteration length, Roles and responsibilities, Virtual team support, Risk mitigation level, Customer interaction and Pros and cons.	(De Haaff, 2017)	Solution
Align the organisation	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Allow grassroots level empowerment	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Allow teams to self-organise	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Arrange social events	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Cherish agile communities	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Coach teams as they learn by doing	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Communicate that change is non-negotiable	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Communicate the change intensively	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Concentrate on agile values	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Conform to a single approach	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Create and communicate positive experiences in the beginning	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Customise the agile approach carefully	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Educate management on agile	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Engage everyone in the organisation	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Ensure management support	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Gather insights from a pilot	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Include persons with previous agile experience	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Invest in learning to refine the requirements	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Keep it simple	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Make management support visible	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Make the change transparent	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor

Map to the old way of working to ease adaptation	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
(Pitkänen, 2015)	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Provide training on agile methods	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Recognise the importance of change leaders	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Recognise the importance of the Product Owner role	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Show strong commitment	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Start with a pilot to gain acceptance	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
Start with agile supporters	(Dikert, Paasivaara & Lassenius, 2016)	Success Factor
When testing needs (functional or non-functional) overlap between different teams, a separate specialised QA team can be established. These people need to be well trained in their area. Coordination between these and development teams need to be defined.	(Dikert, Paasivaara & Lassenius, 2016)	Solution
If multiple releases are followed, both long and short-term planning is done continuously. Otherwise, long-term planning is done in the initiation/inception/discovery phase.	(Fitzgerald & Stol, 2014)	Solution
Changing mindset takes time and managers should be continually mentored and coached.	(Gandomani et al., 2013)	Solution
Coaches should patiently coach the team on the value of the customer in the team.	(Gandomani et al., 2013)	Solution
Companies should use tools that can supply incremental evolution, continuous integration, re-working, version management and support other agile practices.	(Gandomani et al., 2013)	Solution
Documentation should be just enough. Define appropriate knowledge management strategy and distribution of knowledge in different levels of the organisation.	(Gandomani et al., 2013)	Solution
Guide managers to follow a director and coordinator model.	(Gandomani et al., 2013)	Solution
Invest in tools that support distributed agile teams.	(Gandomani et al., 2013)	Solution
Select appropriate personnel and provide the necessary training, mentoring and creating a set of work practices that promote process excellence.	(Gandomani et al., 2013)	Solution
Teach managers to a style of leadership and collaboration.	(Gandomani et al., 2013)	Solution
Changing the Acquisition Process to Support Agile Delivery	(Gorans & Kruchten, 2016)	Success Factor
Conducting “Just Enough” Upfront Work Before the Start of the Agile Project	(Gorans & Kruchten, 2016)	Success Factor

Implementing More Verbal Communication and Dashboards	(Gorans & Kruchten, 2016)	Success Factor
Implementing Reviews that Support Agile Delivery	(Gorans & Kruchten, 2016)	Success Factor
Including the Right Product Owner and Mission Subject Matter Experts	(Gorans & Kruchten, 2016)	Success Factor
Integrating critical speciality skills to support Agile teams	(Gorans & Kruchten, 2016)	Success Factor
Integrating Executive Champions and Stakeholders into an Agile Initiative	(Gorans & Kruchten, 2016)	Success Factor
Planning for IT Infrastructure and Tooling Needs	(Gorans & Kruchten, 2016)	Success Factor
Selecting Top Staff for Lead Roles in the Agile Project	(Gorans & Kruchten, 2016)	Success Factor
Using Existing Knowledge and Not Reinventing the Wheel	(Gorans & Kruchten, 2016)	Success Factor
A pilot project must be selected when first adopting agile, and it should be near the middle of what is the average for an organisation, small enough to be done by one team and should not be critical to the organisation	(Hajjdiab & Taleb, 2011b)	Solution
Careful planning and discipline of agile adoption are required. A team must also expect a slight decrease in productivity when first adopting agile while the teams learn the new implementation techniques.	(Hajjdiab & Taleb, 2011b)	Solution
In agile software engineering literature, the mature Agile coach has a critical role to play to ensure successful outcomes. An organisation needs to hire an Agile coach for at least six months to increases the chance of succeeding in the Agile adoption process.	(Hajjdiab & Taleb, 2011b)	Solution
Many organisations complain about the lack of documentation, but also over-documenting and spending plenty of time documenting unnecessary information is incorrect. What agile development offers is a new way of comprehensive documentation that takes less time and effort. This is another issue that needs to be supported by the upper management and agreed upon with the customers from the beginning of the project.	(Hajjdiab & Taleb, 2011b)	Solution
One of the valuable benefits of agile development is its ability to be customised, based on the culture and the environment of the organisation it is adopted in. These changes should not compromise its beneficiary and affect its efficiency. It might not be realistic to change the whole organisation system overnight, but that does not mean that governmental systems should not be revised and enhanced to what is best for the governmental organisation.	(Hajjdiab & Taleb, 2011b)	Solution
The adoption of agile should be planned in a time that has the minimum work pressure, for instance, a company should invest more time by not accepting new projects for six months and investing this time only for agile adoption.	(Hajjdiab & Taleb, 2011b)	Solution

Upper management approval is fundamental to support any major change in the processes of any department; without their approval, the success of the adoption is doubted. To acquire the upper management support a presentation or a meeting might be conducted to demonstrate the new agile development and explain the benefits of changing to a new method.	(Hajjdiab & Taleb, 2011b)	Solution
Explain the value of coherent, self-organising teamwork.	(Kakar, 2017)	Solution
Agile Champion	(Kropp, Martin & Meier, 2015)	Success Factor
Clean Code	(Kropp, Martin & Meier, 2015)	Success Factor
Collaboration and Communication	(Kropp, Martin & Meier, 2015)	Success Factor
Continuous Integration	(Kropp, Martin & Meier, 2015)	Success Factor
Craftsmanship	(Kropp, Martin & Meier, 2015)	Success Factor
Customers and Requirements	(Kropp, Martin & Meier, 2015)	Success Factor
Organisational Culture	(Kropp, Martin & Meier, 2015)	Success Factor
Testing	(Kropp, Martin & Meier, 2015)	Success Factor
Transparency and Openness	(Kropp, Martin & Meier, 2015)	Success Factor
Focus on the XP core practice of whole team focus and educate on the value of it.	(Lindstrom & Jeffries, 2004)	Solution
While delivering in short release cycles ensure that non-software areas (e.g. marketing) are up to date on progress, in order not to cause bottlenecks and release disappointments. All stakeholders on project need to be Agile, not only software development.	(Lindstrom & Jeffries, 2004)	Solution
Committing to only what is sure and instead pull in more work when all work is done early.	(Marchenko & Abrahamsson, 2008)	Solution
Educate management on self-organisation and that the customer needs are critical to the team's success. They will not ignore it. Management will have to trust them.	(Marchenko & Abrahamsson, 2008)	Solution
It is imperative that the whole need to work well together. Every team requires a mature Scrum Master or Agile leader that will help to facilitate that. All team members' work should be evident.	(Marchenko & Abrahamsson, 2008)	Solution
Communication and negotiation	(Misra, Kumar & Kumar, 2009)	Success Factor
Competency	(Misra, Kumar & Kumar, 2009)	Success Factor
Corporate culture	(Misra, Kumar & Kumar, 2009)	Success Factor

Customer collaboration	(Misra, Kumar & Kumar, 2009)	Success Factor
Customer commitment	(Misra, Kumar & Kumar, 2009)	Success Factor
Customer satisfaction	(Misra, Kumar & Kumar, 2009)	Success Factor
Decision time	(Misra, Kumar & Kumar, 2009)	Success Factor
Personal characteristics	(Misra, Kumar & Kumar, 2009)	Success Factor
Planning and control	(Misra, Kumar & Kumar, 2009)	Success Factor
Societal culture	(Misra, Kumar & Kumar, 2009)	Success Factor
Team distribution	(Misra, Kumar & Kumar, 2009)	Success Factor
Team size	(Misra, Kumar & Kumar, 2009)	Success Factor
Training and learning	(Misra, Kumar & Kumar, 2009)	Success Factor
Coherent, self-organising teamwork	(Nguyen, 2016)	Success Factor
The collocation of the whole team	(Nguyen, 2016)	Success Factor
Committed sponsor	(Nguyen, 2016)	Success Factor
Competency	(Nguyen, 2016)	Success Factor
Cooperative organisational culture instead of hierarchal	(Nguyen, 2016)	Success Factor
Corporate Culture	(Nguyen, 2016)	Success Factor
Cultural training and facilitation aiming at cultural differences in media utilisation and communication could also evidence beneficial for Agile software development functioning.	(Nguyen, 2016)	Solution
Customer commitment	(Nguyen, 2016)	Success Factor
Customer having full authority	(Nguyen, 2016)	Success Factor
Development	(Nguyen, 2016)	Success Factor
Education and Training	(Nguyen, 2016)	Success Factor
Facility with the proper agile-style work environment	(Nguyen, 2016)	Success Factor
Following agile-oriented configuration management process;	(Nguyen, 2016)	Success Factor
Following agile-oriented project management process	(Nguyen, 2016)	Success Factor
Following agile-oriented requirement management process;	(Nguyen, 2016)	Success Factor
Good customer relationship	(Nguyen, 2016)	Success Factor
Honouring regular working schedule – no overtime	(Nguyen, 2016)	Success Factor
Managers knowledgeable in the agile process	(Nguyen, 2016)	Success Factor
Managers who have light-touch or adaptive	(Nguyen, 2016)	Success Factor
Motivated team members	(Nguyen, 2016)	Success Factor
Oral culture placing a high value on	(Nguyen, 2016)	Success Factor
Oral culture placing a high value on face to face	(Nguyen, 2016)	Success Factor
Organisations where the agile methodology is universally accepted	(Nguyen, 2016)	Success Factor
Personal characteristics	(Nguyen, 2016)	Success Factor
Planning and Control	(Nguyen, 2016)	Success Factor

Requirements	(Nguyen, 2016)	Success Factor
Reward system appropriate for agile	(Nguyen, 2016)	Success Factor
Strong communication focus with daily face-to-face meetings	(Nguyen, 2016)	Success Factor
Strong customer commitment and presence	(Nguyen, 2016)	Success Factor
Strong executive support;	(Nguyen, 2016)	Success Factor
Team Distribution	(Nguyen, 2016)	Success Factor
Technologies and Development Tools	(Nguyen, 2016)	Success Factor
Testing	(Nguyen, 2016)	Success Factor
Setup more closely co-located project teams and explaining the value of it.	Nivoit, 2013	Solution
Educate the organisation on the value of Agile coaching and the risks of not doing it.	(O'Connor & Duchonova, 2014)	Solution
Change leadership	(Pitkänen, 2015)	Success Factor
Coaching	(Pitkänen, 2015)	Success Factor
Communities of practices	(Pitkänen, 2015)	Success Factor
Focus on getting the product owner role right	(Pitkänen, 2015)	Success Factor
Focus on getting the product owner role right	(Pitkänen, 2015)	Success Factor
Pragmatic customisation of the agile process	(Pitkänen, 2015)	Success Factor
A structured approach to facilitate inter-team communication	(Pitkänen, 2015)	Success Factor
Synchronizing practices of teams	(Pitkänen, 2015)	Success Factor
Training	(Pitkänen, 2015)	Success Factor
Using pilot projects to test and evaluate changes in process	(Pitkänen, 2015)	Success Factor
High-level requirements upfront in Agile are not missing, but it somewhat lacks detail in the initial state. Explain that Detailed requirements will be developed during the development phase.	(Ramesh, Cao & Baskerville, 2010)	Solution
Waterfall gives a perception of predictability, but as it is mostly unempirical, it is less predictable. Empirical, incremental and iterative Agile models utilising velocity provides more predictability.	(Schwaber, Laganza & D'Silva, 2007)	Solution
Agile teams must frequently communicate with senior management to clearly articulate their successes and challenges, and they should share their experiences with the broader organisation, so their achievements are acknowledged.	(Scrumology, 2009)	Solution
Educate management and business partners on a single approach and the value of it.	(Scrumology, 2009)	Solution
Encourage team members to own and solve their problems instead of letting organisations control technical conversations.	(Scrumology, 2009)	Solution
Every member of the team needs to contribute, and if one or two members of the team are putting themselves above the team, this needs to be quickly addressed by management. Often, this results in the quick removal of the individual(s).	(Scrumology, 2009)	Solution
If the Scrum Master is ineffective at protecting the team, this person needs to be replaced. To find a new Scrum Master, request volunteers or ask the team whom they would like to represent them.	(Scrumology, 2009)	Solution

Implementing Test-Driven Development (TDD) and Continuous Integration (CI) are reasonable first steps. The tools for both of these practices are standard and (often) free.	(Scrumology, 2009)	Solution
In the situation where a single customer cannot be identified, pinpoint the project sponsor, who is the single person who ultimately approves the funding of the project. By working with the sponsor and clearly articulating the need for a single business representative on the project team, this situation can usually be quickly resolved.	(Scrumology, 2009)	Solution
In the situation where several different groups have an equal interest in the success of a project, the team still needs a single representative who is willing to work with each of the different groups and prioritise accordingly. Again, work with the project sponsor to provide the path to a solution.	(Scrumology, 2009)	Solution
Let the teamwork directly with the customer	(Scrumology, 2009)	Solution
Long-term solutions are dependent on rewarding teamwork and breadth of understanding. While there is no single or easy solution, some ideas include having team goals and bonuses, eliminating individual performance reviews, and allowing teams the opportunity to make mistakes.	(Scrumology, 2009)	Solution
Management requires education, coaxing and convincing to recognise that there is a better way to develop software.	(Scrumology, 2009)	Solution
Motivate, educate and train teams in the use of Test-driven development (TDD), Pair programming, continuous integration (CI) and refactoring as starters.	(Scrumology, 2009)	Solution
Solicit input from different members of the team.	(Scrumology, 2009)	Solution
The first step in educating an organisation about Agile development is speaking to the team or individual members of the team. However, you also need to consider a wider audience, including functional managers; the PMO and HR. Failure to address this wider audience can hobble the transition to Agile.	(Scrumology, 2009)	Solution
The Scrum Master and project team need to address the issue with senior management. They need to explain why specific tools are relevant and need to be supported, and they need to make management understand that having the right tools can make the difference between a good enough product and a high-quality product.	(Scrumology, 2009)	Solution
The standard approach is to prioritise dependant functionality early and to code to an agreed-upon the interface, but this is overly simplistic. Instead, teams need to recognise that there is a dynamic relationship between dependant teams that need to be actively and continuously improved.	(Scrumology, 2009)	Solution
To minimise hand-offs, encourage pairing at all stages and ensure that the necessary handoffs are informal. Encouraging the team to tackle small chunks of work also helps because it forces them to communicate more frequently.	(Scrumology, 2009)	Solution

When working with management, pay attention to both Agile metrics (or the lack, thereof), and adaptive planning over predictive planning. The Agile approach to both of these is counter-intuitive for most classically trained managers and requires constant reinforcing.	(Scrumology, 2009)	Solution
Utilising a maturity and risk assessment on the organisation will help to determine the main resistance criteria. Tools like the AAC, 4 stage process (identifying discontinuing factors) and SAMI of Sidky (2007) might be useful.	(Sidky, 2007)	Solution
In Agile, "Just in Time" and "Sprint Pairs" are the most common UX models. Just in Time - (1. Design done within a sprint. 2. Typically requires predefined and commonly used chunks of design. 3. Requires much collaboration. 4. Iterative with parallel efforts and predicted throwaway work. 5. Easier to track.) Also, Sprint Pairs - (1. Design works a sprint ahead of development. 2. The scope is traded off for designing within the time box. 3. Respects functional dependencies.)	(Sintes, 2017)	Solution

Table 6.2: Unnormalised list of Agile adoption success factors and solutions

Id	challenge	origin	categ.	severity	difficult	priority
4	Developers fear their skill deficiencies will be more transparent.	1	1	3	2.04	2.47
5	There is a perceived need for developers to be a "master of all trades."	1	1	2.46	2.53	1.62
6	There is an increased reliance on social skills. E.g. Pair Programming and deeper need for collaboration.	1	1	2.82	2.77	1.9
7	Lack of business knowledge among developers.	1	1	3.76	1.7	4.04
8	Practices are prioritized over the Agile values and principles, and we do not understand it well enough.	1	1	3.27	2.08	2.81
9	Lack of developer motivation to use agile methods (Developer resistance).	1	1	3.4	2.1	2.97
10	The implications of devolved decision-making (Team making decisions).	1	1	4.73	3.92	3.48
11	There is a need for agile-compliant performance evaluation.	5	3	4.35	4.47	2.86
12	Lack of agile-specific recruitment policies and suitably trained IT graduates.	1	2	2.61	2.75	1.7
13	There is a general organizational resistance to change.	4	1	4	4	2.67
14	Lack of management support.	2	1	4.71	3.89	3.47
15	Lack of training.	3	2	4	1.56	4.75
16	Lack of Peer Support.	5	1	3.3	2.08	2.85
17	Lack of formal guidelines on Agile, especially Quality, Scaling and practices.	3	3	3.67	2.78	2.84
18	The perceived rewards for Using Agile Techniques are minimal.	1	3	3.06	2.12	2.49
19	There is scepticism towards the new way of working.	5	1	3	3	2
20	There is a lack of change management during the adoption process	3	3	4.35	3.66	3.17
21	Management is unwilling to change (waterfall mode; Command and Control).	2	1	3.88	4.54	2.4
22	Lack of coaching and mentorship.	3	2	3.84	2.39	3.34

23	There is a perception of too high a workload. Balancing an Agile project with other traditional projects is difficult.	3	3	2.92	2	2.4
24	Co-location challenges are hard to resolve.	4	2	2.37	1.89	1.78
25	Previous Agile attempts were customized poorly.	3	3	2.88	3.04	1.87
26	Certain teams and individuals are overzealous	3	3	2	1.86	1.38
27	Coordination and management of work across teams are hard.	3	4	4.24	3.8	2.98
28	The autonomous team model is challenging (self-organization).	4	4	3.57	4.22	2.2
29	Communication over distributed teams is challenging.	4	4	3.16	3.04	2.15
30	Integration and technical consistency are very hard to achieve over multi-interdependent areas.	6	7	4.08	4.24	2.67
31	Middle managers do not know what role they play in Agile.	5	6	3.59	3.53	2.41
32	Company philosophy or culture at odds with core agile values. Bureaucratic with too much governance and compliance.	4	1	4	4.39	2.55
33	Internal silos are kept.	4	6	4.4	4.24	2.99
34	There is a perception that high-level requirements management is largely missing in Agile.	3	7	2.31	2	1.66
35	Requirements refinement are challenging.	3	7	2.63	2.8	1.7
36	The creation and estimation of user stories are hard.	3	3	2.71	2.43	1.91
37	The perceived gap between long and short-term planning	3	7	1.71	1.38	1.27
38	Nonfunctional testing (performance security, scalability, etc.) is hard for large projects.	3	8	3	3.3	1.91
39	Lack of automated testing.	3	8	3.97	3.36	2.89
40	There is a perception that requirements are conflicting over multiple teams and sprints. This has a severe impact on QA.	3	7	2.25	2	1.6
41	Other functions of the company are unwilling to change.	4	9	3.63	3.12	2.62
42	Short, iterative and incremental deliveries are a challenge.	3	3	2.61	2.71	1.71
43	Product launch activities cannot be easily adjusted.	3	9	2.18	2.98	1.26
44	Traditional need to overgenerate documentation. There is a perception that this is a safety net.	3	3	2.46	2.53	1.62
45	Cultural differences in multi-international teams.	5	4	3.38	2.63	2.58
46	Customer or client requirement to be part of the team.	5	9	3.94	3.57	2.76
47	The complexity of moving from a traditional to an Agile approach.	4	3	2.67	3.27	1.61
48	Choosing an appropriate Agile method is hard.	6	3	1.96	1.71	1.4
49	Using non-flexible tools and hardware is a barrier in moving to agile.	7	2	3.1	2.41	2.36
50	Existing tools are inappropriate.	7	2	3.69	2.98	2.76
51	Team members, in general, find it difficult to work effectively in a team.	5	3	3.69	3.3	2.6
52	The perception that Agile demands a high level of competence.	5	1	2.61	2.45	1.79
53	The uncertainty that Agile can scale properly for large projects.	6	3	3.71	3.58	2.51
54	Specialized skills are needed; like refactoring, configuration management, developer unit testing (TDD), etc.	5	1	3.35	4	2.05
55	Team members find it difficult to change the roles they are used to.	5	1	4.25	3.69	3.06
56	Lack of previous experience with agile methods.	5	3	2.29	2.2	1.55

57	The role of a single identifiable Product Owner is missing.	5	3	3.92	3.29	2.86
58	Collaboration in and between teams as well as with stakeholders are ineffective.	5	3	4.24	3.33	3.21
59	A pilot project was not used.	3	3	1.84	1.45	1.39
60	Requirements are difficult to identify and to have proper traceability to end product.	3	7	2.26	2.53	1.43
61	Measuring Agile success is difficult.	3	1	3.56	3.29	2.47
62	Defining business value is difficult.	3	7	3.98	3.37	2.89
63	Tailoring the appropriate Agile/Lean practices are difficult.	3	3	2.14	2.37	1.35
64	Lack of big design upfront.	1	3	2.24	3	1.3
65	Contracts are fixed priced.	4	3	2.22	2.51	1.39
66	Steep learning curve.	5	3	2.25	2.3	1.49
67	The perception that predictability has decreased in Agile.	3	3	3.52	2.51	2.82
68	Lack of understanding of how to include user-centred design people and processes in large-scale Agile projects.	3	3	3.22	3.93	1.95
69	Requirements management with large/complex products is difficult.	3	7	2.73	3.53	1.61
70	Continuous testing surfaces a lot of practical issues.	3	3	2.9	2.8	1.97
71	The perception that maintenance effort has increased with an increase in the number of releases -- undermining the team productivity and morale.	3	3	3.69	3.2	2.65
72	Requirements priority lists are hard to create and maintain.	3	7	3.16	2.78	2.25
73	Interdependency management between various systems is hard and adds to the complexity of the architecture.	7	7	3.76	4.57	2.29
74	Variance on the speed of delivery between Agile and traditional running concurrently and having interdependencies.	3	3	3.22	3.18	2.16
75	Legacy systems do not integrate with Agile methods easily.	3	7	3.98	3.73	2.74
76	Techniques and reporting in Agile to measure progress is difficult.	7	3	3	3.14	1.96
77	Process standard ratings like ISO and CMMI levels are impacted.	4	3	2.18	1.56	1.74
78	Teams are split, and critical resources are removed after a successful pilot.	3	3	4.44	2.98	3.69
79	Team members are overly-specialized, which requires work to be handed off several times.	3	3	3.41	2.9	2.47
80	Lack of ownership by the team.	5	3	4.3	3.29	3.3
81	It can be difficult to convince management of the need for a new development approach.	2	3	2.33	1.56	1.94
82	Management wants to combine elements of RUP and Agile.	2	3	1.64	1	1.44
83	The ScrumMaster refuses to protect the team.	5	1	1.92	1.4	1.52
84	There is not a proper continuous integration or build system and process in place.	7	3	3.57	3.16	2.53
85	Agile quality concepts are not in place and not well understood.	3	3	3.14	2.53	2.35
86	Possible frictions between teams due to various reasons.	5	4	2.2	2.14	1.48
87	The focus is more on the individuals than the team (Over-individualism).	5	3	3.08	3.27	1.99
88	Teams are committing to too much.	3	3	2.67	2.02	2.06
89	Management is interfering too much.	2	1	2.9	2.22	2.23
90	Legislation; the rules, procedures and external audits are stifling the flow.	4	6	3.46	3.52	2.29

91	Individual and team resistance to change. The main reason is the fear of losing a job (Gandomani et al., 2014).	2	1	2.78	2.53	1.95
92	Availability and time consumption of training courses are worrisome.	5	2	3	2.45	2.22
93	The identification of true customer needs is difficult.	5	7	3.69	3.67	2.46
94	The customer can introduce changes as much as they want.	5	7	3.57	2	3.31
95	Coordination complexities between multiple product owners in large-scale projects.	7	3	4.55	3.61	3.42
96	Issue tracking and traceability in Agile is difficult.	7	3	1.78	1.72	1.21
97	Inherent emphasis on non-functional requirements.	6	9	2	1.76	1.42
98	Quality assurance requirement of multiple reviews and inspections.	6	3	2.08	1.67	1.56
99	Agile Risk management with respect to identification, prioritization and management is not clear.	6	3	2.73	2.22	2.03
100	Tracking velocity in a scaled environment is not clear.	6	3	2.04	2.02	1.37

Table 6.3: Agile Challenges from the database

8.2 Ethical Clearance

UNISA university of south africa

UNISA COLLEGE OF SCIENCE, ENGINEERING AND TECHNOLOGY'S (CSET) RESEARCH AND ETHICS COMMITTEE

20 August 2018

Ref #: 039/EJS/2018/CSET_SOC
Name: Mr Emil Johann Schnabel
Student #: 32192770

Dear Mr Emil Johann Schnabel

Decision: Ethics Approval for 3 years (Humans involved)

Researchers: Mr Emil Johann Schnabel, 16 Esdoring Crescent, Vredekloof, Brackenfell, 7560, 32192770@mylife.unisa.ac.za, +27 84 682 7776

Project Leader(s): Prof Ernest Mnkandla, mnkane@unisa.ac.za, +27 11 670 9059

Working Title of Research:

An automated framework to support Agile Adoption Coaching

Qualification: MSc in Computing

Thank you for the application for research ethics clearance by the Unisa College of Science, Engineering and Technology's (CSET) Research and Ethics Committee for the above mentioned research. Ethics approval is granted for a period of three years, from 20 August 2018 to 20 August 2021.

1. The researcher will ensure that the research project adheres to the values and principles expressed in the UNISA Policy on Research Ethics.
2. Any adverse circumstance arising in the undertaking of the research project that is relevant to the ethicality of the study, as well as changes in the methodology, should be communicated in writing to the Unisa College of Science, Engineering and Technology's (CSET) Research and Ethics Committee. An amended application could be requested if there are substantial changes from the existing proposal, especially if those changes affect any of the study-related risks for the research participants.

Open Rubric

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za

The researcher(s) will conduct the study according to the methods and procedures set out in the approved application.

3. Any changes that can affect the study-related risks for the research participants, particularly in terms of assurances made with regards to the protection of participants' privacy and the confidentiality of the data, should be reported to the Committee in writing, accompanied by a progress report.
4. The researcher will ensure that the research project adheres to any applicable national legislation, professional codes of conduct, institutional guidelines and scientific standards relevant to the specific field of study. Adherence to the following South African legislation is important, if applicable: Protection of Personal Information Act, no 4 of 2013; Children's act no 38 of 2005 and the National Health Act, no 61 of 2003.
5. Only de-identified research data may be used for secondary research purposes in future on condition that the research objectives are similar to those of the original research. Secondary use of identifiable human research data requires additional ethics clearance.
6. No field work activities may continue after the expiry date (20 August 2021). Submission of a completed research ethics progress report will constitute an application for renewal of Ethics Research Committee approval.

Note:

The reference number 039/EJS/2018/CSET_SOC should be clearly indicated on all forms of communication with the intended research participants, as well as with the Unisa College of Science, Engineering and Technology's (CSET) Research and Ethics Committee.

Yours sincerely

Dr. B Chimbo

Chair: Ethics Sub-Committee SoC, College of Science, Engineering and Technology (CSET)

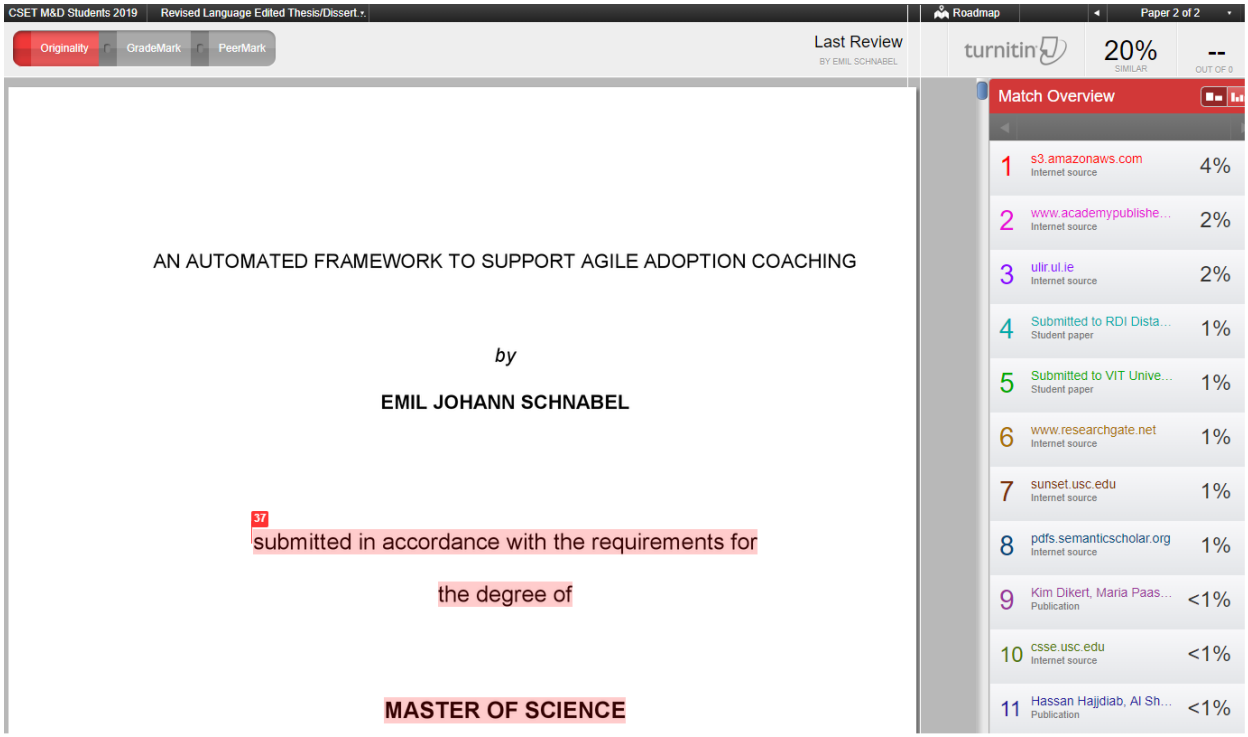
Prof I. Osunmakinde

Director: School of Computing, CSET

Prof B. Mamba

Executive Dean: CSET

8.3 Turnitin Report



8.4 Proof of Professional Language Editing

27 January 2019

To whom it may concern

CERTIFICATION OF EDITING

I hereby state that I proofread and edited the M.Sc. Dissertation, *An Automated Framework to Support Agile Adoption Coaching*, of Mr Emil Schnabel.

Please feel free to contact me with regards to this matter.

Kind Regards

Luan Staphorst

Freelance Proofreader, Editor and Translator

Tel: +27 76 379 7472

Email: luanstaphorst@gmail.com

8.5 Questionnaire Questions for Surveys

8.5.1 Priority Survey

The Agile adoption challenges priority survey posed the questions in the screenshots below. A Likert scale was used to rate both the severity and difficulty of challenges.

✚ Complete Columns "Severity" And "Difficulty"

Challenges priority

Likert #	Severity	Difficulty
1	No noticeable impact on the business.	No difficulty at all in resolving the challenge.
2	Minimal impact on business.	Somewhat difficult to resolve.
3	Noticeable impact on business.	Difficult to resolve the challenge
4	Big impact on business.	Very difficult to resolve the challenge.
5	Major impact on business.	Extremely difficult to resolve.

Please select an option in the columns "Severity" and "Difficulty" for each challenge below. Please refer to the Likert scale in the table above.

Search:

Show 100 entries

#	Challenge	Severity	Difficulty
1	Agile quality concepts are not in place and not well understood.	1 2 3 4 5	1 2 3 4 5
2	Agile Risk management with respect to identification, prioritization and management is not clear.	1 2 3 4 5	1 2 3 4 5
3	Availability and time consumption of training courses are worrisome.	1 2 3 4 5	1 2 3 4 5
4	Certain teams and individuals are overzealous in their Agile approach.	1 2 3 4 5	1 2 3 4 5
5	Choosing an appropriate Agile method is hard.	1 2 3 4 5	1 2 3 4 5
6	Co-location challenges are hard to resolve.	1 2 3 4 5	1 2 3 4 5
7	Collaboration in and between teams as well as with stakeholders are ineffective.	1 2 3 4 5	1 2 3 4 5
8	Communication over distributed teams is challenging.	1 2 3 4 5	1 2 3 4 5
9	Company philosophy or culture at odds with core agile . Bureaucratic with too much governance and compliance.	1 2 3 4 5	1 2 3 4 5
10	Continuous testing surfaces a lot of practical issues.	1 2 3 4 5	1 2 3 4 5
11	Contracts are fixed priced.	1 2 3 4 5	1 2 3 4 5
12	Coordination and management of work across teams are hard.	1 2 3 4 5	1 2 3 4 5
13	Coordination complexities between multiple product owners in large-scale projects.	1 2 3 4 5	1 2 3 4 5
14	Cultural differences in multi-international teams.	1 2 3 4 5	1 2 3 4 5
15	Customer or client requirement to be part of the team.	1 2 3 4 5	1 2 3 4 5
16	Defining business value is difficult.	1 2 3 4 5	1 2 3 4 5
17	Developers fear their skill deficiencies will be more transparent.	1 2 3 4 5	1 2 3 4 5
18	Existing tools are inappropriate.	1 2 3 4 5	1 2 3 4 5
19	Individual and team resistance to change. The main reason is fear of losing a job (Gandomani et al. 2014).	1 2 3 4 5	1 2 3 4 5
20	Inherent emphasis on non-functional requirements.	1 2 3 4 5	1 2 3 4 5
21	Integration and technical consistency are very hard to achieve over multi-interdependent areas.	1 2 3 4 5	1 2 3 4 5
22	Interdependency management between various systems is hard and adds to the complexity of the architecture.	1 2 3 4 5	1 2 3 4 5
23	Internal silos are kept.	1 2 3 4 5	1 2 3 4 5
24	Issue tracking and traceability in Agile is difficult.	1 2 3 4 5	1 2 3 4 5
25	It can be difficult to convince management for the need for a new development approach.	1 2 3 4 5	1 2 3 4 5
26	Lack of agile-specific recruitment policies and suitably trained IT graduates.	1 2 3 4 5	1 2 3 4 5
27	Lack of automated testing.	1 2 3 4 5	1 2 3 4 5
28	Lack of big design upfront.	1 2 3 4 5	1 2 3 4 5
29	Lack of business knowledge among developers.	1 2 3 4 5	1 2 3 4 5
30	Lack of coaching and mentorship.	1 2 3 4 5	1 2 3 4 5
31	Lack of developer motivation to use agile methods (Developer resistance).	1 2 3 4 5	1 2 3 4 5
32	Lack of formal guidelines on Agile especially Quality, Scaling and practices.	1 2 3 4 5	1 2 3 4 5
33	Lack of management support.	1 2 3 4 5	1 2 3 4 5
34	Lack of ownership by the team.	1 2 3 4 5	1 2 3 4 5
35	Lack of Peer Support.	1 2 3 4 5	1 2 3 4 5
36	Lack of previous experience with agile methods.	1 2 3 4 5	1 2 3 4 5
37	Lack of training.	1 2 3 4 5	1 2 3 4 5
38	Lack of understanding on how to include user-centered design people and processes in large-scale Agile projects.	1 2 3 4 5	1 2 3 4 5
39	Legacy systems do not integrate with Agile methods easily.	1 2 3 4 5	1 2 3 4 5
40	Legislation; the rules, procedures and external audits are stifling the flow.	1 2 3 4 5	1 2 3 4 5
41	Management is interfering too much.	1 2 3 4 5	1 2 3 4 5
42	Management is unwilling to change (waterfall mode; Command and Control).	1 2 3 4 5	1 2 3 4 5
43	Management wants to combine elements of RUP and Agile.	1 2 3 4 5	1 2 3 4 5
44	Measuring Agile success is difficult.	1 2 3 4 5	1 2 3 4 5
45	Middle managers do not know what role they play in Agile.	1 2 3 4 5	1 2 3 4 5
46	Nonfunctional testing (performance security, scalability, etc is hard for large projects.	1 2 3 4 5	1 2 3 4 5
47	Other functions of the company are unwilling to change.	1 2 3 4 5	1 2 3 4 5
48	Possible frictions between teams.	1 2 3 4 5	1 2 3 4 5
49	Practices are prioritized over the Agile and principles, and we do not understand it well enough.	1 2 3 4 5	1 2 3 4 5
50	Previous Agile attempts were customized poorly.	1 2 3 4 5	1 2 3 4 5
51	Process standard ratings like ISO and CMMI levels are impacted.	1 2 3 4 5	1 2 3 4 5
52	Product launch activities cannot be easily adjusted.	1 2 3 4 5	1 2 3 4 5
53	Quality assurance requirement of multiple reviews and inspections.	1 2 3 4 5	1 2 3 4 5
54	Requirements are difficult to identify and to have proper traceability to end product.	1 2 3 4 5	1 2 3 4 5
55	Requirements management with large/complex products is difficult.	1 2 3 4 5	1 2 3 4 5

57 Requirements refinement are challenging.	1 2 3 4 5	1 2 3 4 5
58 Short, iterative and incremental deliveries are a challenge.	1 2 3 4 5	1 2 3 4 5
59 Specialized skills are needed; like refactoring, configuration management, developer unit testing (TDD), etc.	1 2 3 4 5	1 2 3 4 5
60 Steep learning curve.	1 2 3 4 5	1 2 3 4 5
61 Tailoring the appropriate Agile/Lean practices are difficult.	1 2 3 4 5	1 2 3 4 5
62 Team members are overly-specialized, which requires work to be handed off several times.	1 2 3 4 5	1 2 3 4 5
63 Team members find it difficult to change the roles they are used to.	1 2 3 4 5	1 2 3 4 5
64 Team members, in general, find it difficult to work effectively in a team.	1 2 3 4 5	1 2 3 4 5
65 Teams are committing to too much.	1 2 3 4 5	1 2 3 4 5
66 Teams are split, and key resources are removed after a successful pilot.	1 2 3 4 5	1 2 3 4 5
67 Techniques and reporting in Agile to measure progress is difficult.	1 2 3 4 5	1 2 3 4 5
68 The absence of a Pilot Project.	1 2 3 4 5	1 2 3 4 5
69 The autonomous team model is challenging (self-organization).	1 2 3 4 5	1 2 3 4 5
70 The complexity of moving from a traditional to an Agile approach.	1 2 3 4 5	1 2 3 4 5
71 The creation and estimation of user stories are hard.	1 2 3 4 5	1 2 3 4 5
72 The customer can introduce changes as much as they want.	1 2 3 4 5	1 2 3 4 5
73 The focus is more on the individuals than the team (Over-individualism).	1 2 3 4 5	1 2 3 4 5
74 The identification of the true customer needs is difficult.	1 2 3 4 5	1 2 3 4 5
75 The implications of devolved decision-making (Team making decisions).	1 2 3 4 5	1 2 3 4 5
76 The perceived gap between long and short-term planning.	1 2 3 4 5	1 2 3 4 5
77 The perceived rewards for Using Agile Techniques are minimal.	1 2 3 4 5	1 2 3 4 5
78 The perception that Agile demands a high level of competence.	1 2 3 4 5	1 2 3 4 5
79 The perception that maintenance effort has increased with an increase in the number of releases; undermining the team productivity and morale.	1 2 3 4 5	1 2 3 4 5
80 The perception that predictability has decreased in Agile.	1 2 3 4 5	1 2 3 4 5
81 The role of a single identifiable Product Owner is missing.	1 2 3 4 5	1 2 3 4 5
82 The ScrumMaster refuses to protect the team.	1 2 3 4 5	1 2 3 4 5
83 The uncertainty that Agile can scale properly for large projects.	1 2 3 4 5	1 2 3 4 5
84 There is a general organizational resistance to change.	1 2 3 4 5	1 2 3 4 5
85 There is a lack of change management during the adoption process.	1 2 3 4 5	1 2 3 4 5
86 There is a need for agile-compliant performance evaluation.	1 2 3 4 5	1 2 3 4 5
87 There is a perceived need for developers to be a master of all trades.	1 2 3 4 5	1 2 3 4 5
88 There is a perception of too high workload. Balancing an Agile project with other traditional projects is difficult.	1 2 3 4 5	1 2 3 4 5
89 There is a perception that high-level requirements management is largely missing in Agile.	1 2 3 4 5	1 2 3 4 5
90 There is a perception that requirements are conflicting over multiple teams and sprints. This have a severe impact on QA.	1 2 3 4 5	1 2 3 4 5
91 There is an increased reliance on social skills. Eg. Pair Programming and deeper need for collaboration.	1 2 3 4 5	1 2 3 4 5
92 There is not a proper continuous integration or build system and process in place.	1 2 3 4 5	1 2 3 4 5
93 There is scepticism towards the new way of working.	1 2 3 4 5	1 2 3 4 5
94 Tracking velocity in a scaled environment is not clear.	1 2 3 4 5	1 2 3 4 5
95 Traditional need to overgenerate documentation. There is a perception that this is a safety net.	1 2 3 4 5	1 2 3 4 5
96 Using non-flexible tools and hardware is a barrier in moving to agile.	1 2 3 4 5	1 2 3 4 5
97 Variance on the speed of delivery between Agile and traditional running concurrently and having interdependencies.	1 2 3 4 5	1 2 3 4 5

Showing 1 to 97 of 97 entries

First Previous 1 Next Last

8.5.2 **AAC Survey**

The AAC questionnaire below was posed after a respondent has completed an AAC exercise from the following URL:

http://www.agilesense.co.za/aacassessment/#/home